ЗАТВЕРДЖЕНО

Наказ ДВНЗ "ПДТУ"

від 16.04.2014 року № 75-05

**Форма ПДТУ – 3б**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД**

**«ПРИАЗОВСЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»**

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ |
| **КАФЕДРА** | КОМП’ЮТЕРНІ НАУКИ |
| **СПЕЦІАЛЬНІСТЬ** | 6.050101 “КОМП’ЮТЕРНІ НАУКИ” |

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**ДО ДИПЛОМНОЇ РОБОТИ БАКАЛАВРА**

|  |  |
| --- | --- |
| **НА ТЕМУ:** | АВТОМАТИЗОВАНА ІНФОРМАЦІЙНА СИСТЕМА ОБЛІКУ РОБОТИ |
| ВЧИТЕЛЯ-ЛОГОПЕДА ПСИХОЛОГО-МЕДИКО-ПЕДАГОГІЧНОЇ КОНСУЛЬТАЦІЇ | |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| **СТУДЕНТ** |  |  | **/В.С. ХАРАМАН/** |
|  | (підпис) |  | (ПІБ) |
| **КЕРІВНИК РОБОТИ** |  |  | **/Т.О. ЛЕВИЦЬКА/** |
|  | (підпис) |  | (ПІБ) |

**КОНСУЛЬТАНТИ:**

|  |  |  |  |
| --- | --- | --- | --- |
| **З ОХОРОНИ ПРАЦІ** |  |  | **/О.Ю. НЕСТЕРОВ/** |
|  | (підпис) |  | (ПІБ) |
| **З НОРМОКОНТРОЛЮ** |  |  | **/Т.О. ЛЕВИЦЬКА/** |
|  | (підпис) |  | (ПІБ) |

|  |  |  |  |
| --- | --- | --- | --- |
| **РЕЦЕНЗЕНТ** |  |  | **/А.В. ГАДАШЕВИЧ/** |
|  | (підпис) |  | (ПІБ) |

**РОБОТА РОЗГЛЯНУТА НА ЗАСІДАННІ КАФЕДРИ І ДОПУЩЕНА ДО ЗАХИСТУ В ДЕК**

**протокол від «20» травня 2016 р. № 8**

|  |  |  |  |
| --- | --- | --- | --- |
| **ЗАВІДУВАЧ КАФЕДРИ** |  |  | **/І.В. ФЕДОСОВА/** |
|  | (підпис) |  | (ПІБ) |

**МАРІУПОЛЬ 2016 р.**

ЗАТВЕРДЖЕНО

Наказ ДВНЗ "ПДТУ"

від 16.04.2014 року № 75-05

**Форма ПДТУ – 4б**

**ДВНЗ «ПРИАЗОВСЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Факультет | Інформаційних технологій |
| Кафедра | Комп’ютерні науки |
| Освітньо-кваліфікаційний рівень – | Бакалавр |
| Напрям підготовки | 6.050101 «Комп’ютерні науки» |

(шифр і назва)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ЗАТВЕРДЖУЮ | | | | | |
| Зав.кафедри | | | | | Федосова І. В. | |
|  | | | | | |
|  | «15.» |  | 04 | | 2016 р. | |

**З А В Д А Н Н Я**

**НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Харамана Володимира Сергійовича | | | | | | |
| (прізвище, ім’я, по батькові) | | | | | | |
| 1. Тема проекту (роботи) | | Автоматизована інформаційна система обліку роботи | | | | |
| вчителя-логопеда психолого-медико-педагогічної консультації | | | | | | |
|  | | | | | | |
| затверджена наказом по вузу від | | | «03 лютого» | | 2016 р. № | 17-05 |
| 2. Термін здачі студентом закінченого проекту (роботи) | | | | | | 10 червня 2016 р. |
| 3. Вихідні дані до проекту (роботи) | | | | Виписка з протоколу ПМПК, документ | | |
| про мовленнєвий розвиток дитини, рекомендації ПМПК, зауваження батьків | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
|  | | | | | | |
| 4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають | | | | | | |
| розробці: | вступ; аналіз існуючого рівня автоматизації , постановка задачі; | | | | | |
| проектування підсистеми, інформаційне, математичне, організаційне, програмне | | | | | | |
| та технічне забезпечення. | | | | | | |
|  | | | | | | |
|  | | | | | | |
| 5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень): | | | | | | |  |
| слайди мультимедійної презентації | | | | | | |
|  | | | | | | |

6. Консультанти розділів роботи

|  |  |  |  |
| --- | --- | --- | --- |
| зділ | Прізвище, ініціали та посада  консультанта | Підпис, дата | |
| завдання видав | завдання  прийняв |
| Охорона праці | Нестеров О.Ю.,  канд. техн. наук, доц. |  |  |
| Нормоконтроль | Левицька Т.О.,  канд**.** техн. наук, доц. |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| 7. Дата видачі завдання | 15.04.2016 |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| №  з/п | Назва етапів роботи | Строк виконання етапів роботи | Примітка |
| 1 | Аналіз предметної області | 23.04 |  |
| 2 | Постановка задачі | 23.04 |  |
| 3 | Проектування підсистеми | 30.04 |  |
| 4 | Інформаційне та математичне забезпечення | 07.05 |  |
| 5 | Організаційне забезпечення | 21.05 |  |
| 6 | Програмне забезпечення | 28.05 |  |
| 7 | Технічне забезпечення | 04.06 |  |
| 8 | Оформлення пояснювальної записки | 10.06 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Хараман В.С.** |
| (прізвище та ініціали) |

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |
| --- |
| **Левицька Т.О.** |
| (прізвище та ініціали) |

(підпис)

**Керівник роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(підпис)

Реферат

Пояснительная записка содержит 78 с., 45 рисунков, 6 таблиц.

Тема моего дипломного проекта состоит в следующем: разработать автоматизированную информационную систему учёта работы учителя-логопеда психолого-медико-педагогической консультации.

Состоит из: реферата, содержания, введения, специальной части, производственной и экологической безопасности при разработке автоматизированной информационной подсистемы учёта работы учителя-логопеда, экономики и организация производства, выводов, перечня использованных источников и приложений.

Специальная часть включает в себя:

* анализ предметной области (описание предметной области, основные концепции, положенные в основу дипломного проекта, критическая характеристика существующих проектных решений и программного обеспечения существующего класса);
* постановка задачи (наименование программы, назначение и область применения программы, требования к функциональным характеристикам программы, требования к надежности программы, требования к квалификации и численности персонала, требования к технологии хранения и обработки информации проектируемой задачи, требования к системе управления базами данных, требования к составу и параметрам технических средств, требования к исходным кодам и языкам программирования, требования к программным средствам, используемым программой, требования к организации входных и выходных данных, требования к защите информации и программ, предварительный состав программной документации);
* моделирование проектированной задачи (диаграмма вариантов использования, диаграмма базы данных, диаграмма деятельности, диаграмма компонентов и развёртываний);
* программное обеспечение (обоснование выбора СУБД, обоснование выбора языка программирования, обоснование выбора инструментальных возможностей программной реализации задачи, инструкция пользователя);

Выводы включают в себя краткие итоги результатов выполненной работы и достигнутой при этом эффективности, а также затрагиваются вопрос перспективы развития и использования проектных решений.

Перечень использованных источников.

Приложение А – Листинг программы.

КЛЮЧЕВЫЕ СЛОВА: АИС, ЛОГОПЕД, БАЗА ДАННЫХ, СУБД, php, heidiSQL, UML-ДИАГРАММЫ, MYSQL

Содержание

Введение…………………………………….……………………………………………8

1 ОБЩИЙ РАЗДЕЛ …………………….…………...…………………………………9

[1.1 Изучение объекта исследования](#_Toc450119188).............................................................................9

[1.1.1 Краткая информация о предприятии …………………………………………](#_Toc450119189)9

[1.1.2 Подразделения и пользователи системы ………………………………….....](#_Toc450119190)11

[1.1.3 Видение выполнения проекта и границы проекта ………………………......](#_Toc450119191)11

[1.2 Анализ существующего уровня автоматизации…...............................................1](#_Toc450119192)2

[1.3 Общие требования к АИС………………………………………………………...13](#_Toc450119193)

[1.3.1 Функции и бизнес-процессы ………………………………………………….14](#_Toc450119194)

[1.3.2 Матрица организационной ответственности ………………………………...14](#_Toc450119195)

[1.3.3 Оборудование и инструменты ………………………………………………..](#_Toc450119196)15

[1.3.4 Матрица отображения оборудования и инструментов на функции и бизнес-процессы ……………………………………………………………………………..1](#_Toc450119197)5

[1.3.5 Регламентирующие документы ……………………………………………....16](#_Toc450119198)

[1.3.6 Матрица отображения регламентирующих документов на функции и бизнес-процессы …………………………………………………………………….16](#_Toc450119199)

[1.4 Обзор и анализ существующих методов и средств решения задач....................1](#_Toc450119200)7

[1.5 Постановка задачи………………………………………………………………...19](#_Toc450119201)

* + 1. Наименование программы. Назначение и область применения программы…………………………………………………………………………...19
    2. Требования к функциональным характеристикам программы…………....19
    3. Требования к надежности программы………………………………………21
    4. Требования к квалификации и численности персонала……………………21
    5. Требования к технологии хранения и обработки информации проектируемой задачи. Требования к системе управления базами данных…….22
    6. Требования к составу и параметрам технических средств………………...23
    7. Требования к исходным кодам и языкам программирования………....…..23
    8. Требования к программным средствам, используемым программой……..24
    9. Требования к организации входных и выходных данных…………………24
    10. Требования к защите информации и программ…………...……………….24
    11. Предварительный состав программной документации…………………...24

[2. СПЕЦИАЛЬНЫЙ РАЗДЕЛ………………………………………………………....26](#_Toc450119202)

[2.1 Проектирование подсистемы, которая разрабатывается](#_Toc450119203)……………………….26

[2.1.1 Методы и средства структурного системного анализа и проектирования](#_Toc450119204)..26

2.1.2 Методы и средства объектно-ориентированного анализа и проектирования……………………………………………………………………...34

[2.2 Информационное обеспечение](#_Toc450119206)…………………………………………………...40

2.2.1 Концептуальная модель АИС…………………………………………...…....40

[2.2.2 Даталогическая модель АИС ………………………………………………...](#_Toc450119208).41

* + 1. Розробка алгоритмів рішення функціональної задач…...……………….…42

2.3 Программное обеспечение ……………………………………..…………………43

* + 1. Схема взаимодействия программных модулей ………….………………...48
    2. Архитектура системы………..………………………………...…….……….49
       1. Диаграмма компонентов ………………………………….……….……...50

2.3.2.2 Диаграмма развертывания ……………………………………...….………50

* 1. Организационное обеспечение ………………………………………………....51
     1. Инструкция пользователя ……………………………………………..……..51
     2. Инструкция администратора ……………………………………….…...…...60
     3. Другие инструкции (специалиста, программиста, системного программиста)…………………………………………………………….……63
  2. Техническое обеспечение …………………………………….…………………63
     1. Обоснование выбора (разработки) технического обеспечения АИС…….63
     2. Структура сетевой системы (топологическая и логическая)……………..65
     3. Структура комплекса средств автоматизации…………………………..….66
     4. Спецификация оборудовани.………………………………………………...66

3. Охрана труда ……………………………………………………………………….…67

3.1 Опасные, вредные факторы и пожароопасность усовершенствованного технологического процесса при использовании результатов исследования в производстве……………………………………………………………………………..67

3.1.1 Общие сведения о предприятии…………………………………………………..67

3.1.2 Опасные и вредные факторы……………………………………………………...43

3.2 Мероприятия по обеспечению безопасности усовершенствованной технологии (или оборудования) при использовании результатов исследования в производстве.70

* + 1. Освещенность рабочего места…………………………………………………...70
    2. Допустимые значения параметров излучений, генерируемых мониторами… 71
    3. Организация рабочего места в соответствии с требованиями эргономики.......72
    4. Рациональная организация режима труда и отдыха……………………………74

Выводы………………………………………………….……………………..……..…..75

Перечень ссылок………………………………………………………………………....76

Приложение А...................................................................................................................56

Введение

Актуальность работы обусловлена тем, что на данный момент, в большинстве логопедических пунктов, вся информация о детях, расписание, графики работы и прочее — хранится в бумажном виде. Также, на момент выполнения данного проекта, не было обнаружено ни одного программного обеспечения, которое бы удовлетворило запросы учителя-логопеда.

С помощью данной подсистемы, будет улучшен механизм документирования информации в области работы учителя-логопеда, а также механизм предоставления информации для психолого-медико-педагогической консультации. Что, в свою очередь, даст мощный толчок для развития ПМПК и работы учителя-логопеда в сторону современности.

Самым большим преимуществом автоматизированной информационной системы является существенное сокращение затрат времени на рутинные операции и удобство работы.

В первой части пояснительной записки проводится подробный анализ рассматриваемой предметной области. Это позволяет более подробно понять и рассмотреть все аспекты проектируемой системы и избавиться от недостатков. Описываются цели проекта, видение выполнения, функции и бизнес-процессы. Так же описываются преимущества и недостатки разрабатываемой системы.

Во второй части вниманию предоставляются функциональные, информационные и поведенческие модели системы. Модели потоков данных, деревья узлов, генерация базы данных.

В третей части рассматривается охрана труда и все ее аспекты. А также в конце присутствуют выводы, и приложение с исходным кодом программы.

1. ОБЩИЙ РАЗДЕЛ
   1. Изучение объекта исследования

Психолого-медико-педагогические консультации являются методическими учреждениями системы образования и науки Украины, осуществляющими консультативную, методическую, психолого-педагогическую и профилактическую деятельность [4].

В консультации проводится диагностика интеллектуальных способностей ребенка, его психической зрелости и готовности к школьному обучению; выявление индивидуальных особенностей развития, личностных особенностей и возможностей ребенка к обучению; разработка рекомендаций по уточнению, а в необходимых случаях и изменению программы обучения; проведение медицинской диагностики состояния соматического и нервно-психического здоровья; проведение компьютерных энцефалографии и эхо-энцефалографии [7, 13].

* + 1. Краткая информация о предприятии

ПМПК города Мариуполя выполняет функцию городской службы по вопросам определения вида обучения [10, 21]. Деятельность психолого-медико-педагогических консультаций направлена ​​на:

– выявление, психолого-педагогическое изучение, оценку труда и потенциальных возможностей развития детей в возрасте до 18 лет, которые нуждаются в коррекции физического и (или) умственного развития, имеют признаки риска возникновения трудностей познавательной деятельности и поведения;

– подготовку заключения и рекомендаций по развитию, содержанию, форм и методов обучения с учетом особенностей учебно-познавательной деятельности ребенка;

– консультирование родителей (лиц, их заменяющих), педагогических работников, по вопросам выбора возможных форм и методов обучения, в том числе инклюзивного (интегрированного), в сочетании с реабилитационными мероприятиями, социальной адаптации и интеграции в общественную жизнь детей, нуждающихся в коррекции физического и (или) умственного развития;

Учитель-логопед:

– проводит углублённое логопедическое обследование детей для определения структуры и степени выраженности имеющегося дефекта [11];

– комплектует группы для занятий с учётом патологии речи в период с 1 по 15 сентября;

* планирует направления и содержание индивидуальной и групповой коррекционной работы, методической работы на учебный год (с сентября по июнь включительно);

– проводит индивидуальные и групповые занятия по коррекции выявленных нарушений;

– консультирует педагогов, родителей (лиц, их заменяющих) по применению методов и приёмов оказания помощи детям, по результатам коррекции;

– предоставляет председателю ПМПК, отчёт о динамике развития речи детей, имеющих нарушения речи;

– оформляет документы в пределах своей компетенции для представления ребёнка на ПМПК;

– способствует формированию культуры личности, социализации детей.

Оформляет документацию установленного образца:

– планы работы учителя-логопеда;

– карты речевого развития;

– индивидуальные карты развития детей;

– книга анализа результативности работы;

– график работы;

– картотека дидактических игр;

– расписание занятий;

– список детей, зачисленных в ЛП;

– речевой экран [19].

* + 1. Подразделения и пользователи системы

Подразделения системы представлены в таблицах 1.1 и 1.2.

Таблица 1.1 - Подразделение системы "Пользователи"

|  |  |
| --- | --- |
| Пользователи | Их возможности и функции |
| Учитель-логопед | Возможность работать с документацией в электронном виде и редактировать её как можно комфортнее. |
| Заведующая ПМПК | Возможность работать с графиком работы в электронном виде и редактировать ей как можно комфортнее. |
| Администратор | Редактирование и создание профилей\аккаунтов. |

Таблица 1.2 - Подразделение системы "Оборудование"

|  |  |
| --- | --- |
| Оборудование | Назначение |
| ПК | Обеспечение возможности полноценно выполнять учет, кооперирование и выполнение задач в необходимый срок. |

На таблицах отображены пользователи и их возможности, также видно какое необходимо оборудование и для чего именно.

* + 1. Видение выполнения проекта и границы проекта

В рамках проекта рассматривается автоматизация всей документации, приведение её в электронный, удобный для работы вид, которая ранее была в бумажном варианте [20, 22].

Сами занятия и их автоматизация не входит в рамки проекта и не будут задействованы.

Количество пользователей системы: учитель-логопед и заведующая ПМПК. Доступ к материалам и системе запрещен посторонним лицам [17].

В рамках выполнения проекта будут реализовываться такие части системы как:

* web-интерфейс;
* картотека дидактических игр;
* формирование отчетов;
* авторизация;
* документация.
  1. Анализ существующего уровня автоматизации

Список программного обеспечения, которое используется учителями на момент исследования:

– Word и Excel для формирования документаций [5, 24];

– “Ваш логопед”.

Это программные обеспечения помогают развивать ребенка, как дома, так и в руках учителя-логопеда.

Основные функции:

* развивает фонематическое представление у детей, формирует правильное произношение, делению слов на слоги;
* хранения информации о каждом ребёнке;
* комплексная характеристика навыков, полученных им в процессе обучения;
* мониторинг успеваемости и сохранности студентов, отслеживать факты неуспеваемости и своевременно принимать меры.

В таблице 1.3 отображен текущий уровень автоматизации.

Таблица 1.3 – Существующий уровень автоматизации

|  |  |
| --- | --- |
| Объекты автоматизации | Наличие |
| Кол-во рабочих станций | Неограниченно |
| Кол-во сотрудников IT отдела | 4 |
| Кол-во ПК, что одновременно могу работать с | 20 |
| Наличие формы связи | С администрацией |
| Характеристики компьютеров | От Pentium III и выше |
| Операционная система | Windows 2000, XP |

В данной таблице отображены объекты автоматизации (кол-во рабочих станций, сотрудников, ПК и т.д.), и их наличие.

1.3 Общие требования к АИС

Разрабатываемая система должна учитывать следующие основные принципы:

* безопасность;
* удобство в пользовании;
* персонализация.

Система должна быть защищена от несанкционированного доступа к данным извне.

Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм [28]. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям подсистемы [5, 9].

Подсистема должна предоставлять информацию пользователям с учетом персональных настроек пользователей.

* + 1. Функции и бизнес-процессы

В процессе проектирования ИС учитывались следующие бизнес-процессы:

Бизнес-процессы основной деятельности:

– материально-техническое обеспечение деятельности:

1. управление процессом документирования;
2. обеспечение необходимым списком зачисленных детей.

– производственные процессы:

1. составление групп;
2. составление плана;
3. исправления дефектов речи;
4. учет документации.

– обслуживание

1) обеспечение после продажного обслуживания;

Вспомогательные бизнес-процессы:

– инженерно-техническое обеспечение;

– информационное обеспечение;

– документооборот;

– управление персоналом.

* + 1. Матрица организационной ответственности
* *Ответственный***.** Для каждой работы обязательно должен быть указан один и только один ответственный.
* *Исполнители***.** Их может быть несколько, а может быть так, что сам ответственный является и исполнителем. Т.е. в одной ячейке может быть более одного обозначения [12, 16].

В таблице 1.4 представлена матрица ответственности системы.

Таблица 1.4 – Матрица ответственности

|  |  |  |
| --- | --- | --- |
|  | Учитель-логопед | Заведующая ПМПК |
| Управление процессом документирования | О, И |  |
| Составление плана | О, И |  |
| Составление групп | О, И |  |
| Составление графика работы |  | О, И |

Матрица ответственности для Учителя-логопеда и заведующей ПМПК.

* + 1. Оборудование и инструменты

Техническое обеспечение системы должно позволять эффективно решать поставленные задачи. Техническое обеспечение ИС должно быть совместимо с программно-аппаратными средствами [15].

Количественный и качественный состав технических средств должен быть определен на стадии технического проектирования.

Примерные характеристики технического обеспечения:

Минимальная конфигурация для сервера базы данных должна быть: CPU: 16 (32 core); RAM: 32 Gb; HDD: 100 Gb; Network Card: 2 (2 Gbit); Fiber Channel: 4  
Минимальный объем свободного пространства для хранения данных на дисковом массиве должен составлять 1 Тб [6, 8].

* + 1. Матрица отображения оборудования и инструментов на функции и бизнес-процессы

В таблице 1.5 представлена матрица отображения оборудования и инструментов на функции.

Таблица 1.5 - Матрица отображения оборудования и инструментов на функции.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Модуль документации | Модуль расписания | Модуль ПМПК |
| Управление процессом документирования | + |  |  |
| Составление плана | + |  |  |
| Составление групп | + |  |  |
| Составление графика работы |  |  | + |

Матрица отображения оборудования и инструментов на функции. Модули документации, расписания и ПМПК.

* + 1. Регламентирующие документы

В данной системе содержаться следующие регламентирующие документы:

– должностная инструкция учителя-логопеда;

– инструкции ПМПК;

– рабочий план.

* + 1. Матрица отображения регламентирующих документов на функции и бизнес-процессы

В таблице 1.6 представлена матрица отображения регламентирующих документов на функции.

Таблица 1.6 - Матрица отображения регламентирующих документов на функции.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Должностная инструкция учителя-логопеда | Инструкции ПМПК | Рабочий план |
| Управление процессом документирования | + |  |  |
| Составление плана |  |  | + |
| Составление групп | + |  |  |
| Составление графика работы |  | + |  |

Матрица отображения регламентирующих документов на функции. Должностная инструкция учителя-логопеда, ПМПК, и рабочий план.

* 1. Обзор и анализ существующих методов и средств решения задач

Рассмотрим готовые решения, способные заменить разрабатываемое программное обеспечение по теме «Автоматизированная информационная система учёта работы учителя-логопеда психолого-медико-педагогической консультации». Среди них можно отметить: «Играем и учимся», «Ваш логопед».

Программный продукт 1: "Играем и учимся"

Играем и учимся - это программное обеспечение, которое помогает развивать ребёнка как дома, так и в руках учителя-логопеда. А также делает занятия интересней и эффективней.

Основные функции:

* путём игр, развивает фонематическое представление у детей, формирует правильное произношение, обучает звуковому анализу и делению слов на слоги;
* заполнение журнала посещаемости преподавателей и детей;
* взаимодействия с общезаведенческой системой, реализованной с использованием СУБД Oracle 10;

Результаты внедрения:

* мониторинг успеваемости и сохранности студентов, отслеживать факты неуспеваемости и своевременно принимать меры;
* упрощение и уменьшение рутинной работы преподавателей и педагогов.

Программный продукт 2: «Ваш логопед»

Ваш логопед - объектом автоматизации является рабочее место любого учителя-логопеда. Учитель-логопед с помощью данной системы, установленной на персональный компьютер, управляет основными функциями учебной работы.

Основные функции:

* автоматическое составление расписания занятий;
* составление планов учебно-логопедической деятельности.

Результаты внедрения:

* упрощение и уменьшение рутинной работы учителя-логопеда.

Оценка целесообразности разработки нового программного продукта.

Программный продукт 1 ("Играем и учимся") почти удовлетворяет поставленной задаче. Однако функциональность данного программного обеспечения не является максимальной, и множество функций, которые необходимы сейчас для учёта и работы с детьми - отсутствуют. К тому же распространение данного программного продукта является коммерческим.

Программный продукт 2 («Ваш логопед») не полностью соответствует требованиям поставленной задачи. Является конечным вариантом, что говорит о невозможности добавление необходимых функций. Так же данное приложение является устаревшим и не поддерживает современные ОС (Windows 7,8). Данный программный продукт распространяется на коммерческой основе.

Все вышеперечисленные аспекты подтверждают целесообразность разработки нового ПО

* 1. Постановка задачи
     1. Наименование программы. Назначение и область применения программы

Наименование программы «Автоматизированная информационная система «Учёт работы учителя-логопеда»».

Назначение программы. Автоматизированная информация система обеспечит интеграцию интерактивных возможностей коммуникации Учитель-логопед – Заведующая ПМПК, функционала учёта документаций учителя-логопеда, и составление графика работы непосредственно от заведующей ПМПК. Станет удобным инструментом для специалистов в области преподавания логопедии.

Область применения программы. Создаваемая автоматизированная информационная система может применяться учителем-логопедом и заведующей ПМПК с целью использования современного комплексного специализированного инструментария для их коммуникаций.

1.5.2 Требования к функциональным характеристикам программы

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм [1-3]. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования функциональных клавиш, режимов работы, поиска, использования [14, 18]. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы.

Интерфейс должен быть рассчитан на преимущественное использование манипулятора типа «мышь», то есть управление системой должно осуществляться с помощью набора экранных меню, кнопок, значков и т.п. элементов. Клавиатурный режим ввода должен используется главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм.

Планируется реализация системы на русском языке. Все модули системы, надписи экранных форм должны быть представлены на русском языке, в том числе и сообщения, выдаваемые пользователю, за исключением системных сообщений.

Экранные формы должны проектироваться с учетом требований унификации:

– все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

– для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы. Термины, используемые для обозначения типовых операций (добавление информационной сущности, редактирование поля данных), а также последовательности действий пользователя при их выполнении, должны быть унифицированы;

– внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки) будут реализованы одинаково для однотипных элементов;

* интерфейс должен быть прост в понимании;
* назначения элементов интерфейса должно быть интуитивно понятно.

В системе должна быть предусмотрена авторизация. Это будет сделано для того, чтобы не произошло несанкционированного доступа к данным программы. Доступ к системе должен осуществляться с компьютера, подключенного к сети Интернет.

1.5.3 Требования к надежности программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением пользователем совокупности организационно-технических мероприятий, перечень которых приведен ниже:

– организацией бесперебойного питания технических средств;

* использованием лицензионного программного обеспечения;   
  регулярным выполнением рекомендаций о типовых нормах времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств;

– регулярным выполнением требований по защите информации и испытаниях программных средств на наличие компьютерных вирусов.

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

1.5.4 Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — учитель-логопед и системный администратор.

На должность администратора назначается лицо, отвечающее следующим требованиям: профессиональное образование, стаж работы в соответствующей области не менее года.

В перечень задач, выполняемых системным администратором, должно входить:

– поддержания работоспособности технических средств;

* установка и поддержание работоспособности системных программных средств;

– операционной системы;

– установка ПО;

* сохранение работоспособности системы при некорректных операциях пользователя.

Учитель-логопед должен:

– знать все документации, с которыми он должен работать;

– быть опытным пользователем ПК.

Учитель-логопед выполняет следующие должностные обязанности:

– вести учёт документаций, которые будут храниться в базе данных ПМПК;

– анализирует информацию, полученную на основе работы с детьми, и вносит её в базу данных ПМПК через форму программы;

* формирует отчёты и отправляет их заведующему ПМПК.

1.5.5 Требования к технологии хранения и обработки информации проектируемой задачи. Требования к системе управления базами данных

Требования к системе управления базами данных СУБД должна обладать возможностью работы в многопользовательском режиме, восстановления работоспособности при программно-аппаратных сбоях, таких как отключение электропитания, проблемы с аппаратным обеспечением, нештатное завершение работы [21, 23].

С целью обеспечения надежного функционирования в СУБД должно быть предусмотрено:

* сохранение работоспособности системы при некорректных операциях пользователя (ввод некорректных данных);
* сохранение целостности данных при нештатном завершении работы системы;
* резервное копирование данных;
* журналирование операций системы.

Данным требованиям соответствуют такие СУБД, как MSSQL Server, PostgreSQL, MySql, SQLite.

1.5.6 Требования к составу и параметрам технических средств

Требования к серверу: процессор INTEL XEON, минимальный объем оперативной памяти 8Гб, минимальный объем дискового пространства 100гб, операционная системаWindows 2000 Server и выше.

Требования к клиентской машине: процессор IntelXeon 5110 или выше, минимальный объём оперативной памяти 2Гб, минимальный объём дискового пространства на жестком диске 50Гб, операционная система Windows 7.

Для пользователя необходимо интернет-соединения.

1.5.7 Требования к исходным кодам и языкам программирования

Разрабатываемая система должна предусматривать возможность ее дальнейшего развития, модификации и включения новых функций в систему, улучшение кода.

Разрабатываемая система должна представлять собой клиент-серверную архитектуру, база данных которой находится в сети Интернет.

Структура программы должна предусматривать возможность ее развития за счет разработки и включения в нее новых форм и функций.

В качестве языка программирования могут быть выбраны такие языки, как php, ruby. Выбор других языков нецелесообразен.

1.5.8 Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной версией операционной системы Windows 2000 Server или выше.

1.5.9 Требования к организации входных и выходных данных

В процессе работы программы входной информацией для программы должны являться: файлы баз данных, манипуляции мышью, а также информация, вводимая пользователем на клавиатуре ЭВМ, согласно режимам, определяемых выходной экранной информацией.

1.5.10 Требования к защите информации и программ

Вход в программу будет осуществляться путём ввода логина и пароля, чтобы не случилось несанкционированного доступа к данным программы.

1.5.11 Предварительный состав программной документации

Состав программной документации должен включать следующие доку­менты:

* техническое задание (постановку задачи);
* руководство пользователя, содержащее описание всех задокументированных возможностей программы;
* текст программы, содержащий исходный код на одном из указанных языков;
* руководство программиста, содержащее данные о технологии проектирования и программирования данной программы, а также о языке программирования, СУБД, назначении программы;
* список литературы, содержащей теоретический материал, необходимый для создания программы.

1. СПЕЦИАЛЬНЫЙ РАЗДЕЛ
   1. Проектирование подсистемы, которая разрабатывается
      1. Методы и средства структурного системного анализа и проектирования

Краткая характеристика методов и способов.

В данном пункте описаны функциональные, информационные и поведенческие модели системы.

Для функциональных моделей будут применяться методологии:

* IDEF0 – методология функционального моделирования;
* IDEF3 – методология описания процессов;
* DFD – методология моделирования потоков данных;
* IDEF1X – методология моделирования данных.

В качестве ПО для построения диаграмм (Idef0, Idef3, DFD) было выбрано BPwin.

BPwin -  [CASE](https://ru.wikipedia.org/wiki/CASE)-средство для [проектирования](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и [документирования](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D1%8F) [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), которое позволяет создавать, документировать и сопровождать базы данных, хранилища и витрины данных. [Модели данных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) помогают визуализировать структуру данных, обеспечивая эффективный процесс организации, управления и администрирования таких аспектов деятельности предприятия, как уровень сложности данных, технологий баз данных и среды развертывания.

Каждая диаграмма в нотациях IDEF0, IDEF3, DFD предназначена для описания одного или нескольких бизнес-процессов.

В примерах с диаграммой IDEF0, будет использоваться модель AS-IS.

FEO (For Exposition Only) диаграммы (другое название – диаграммы только для экспозиции, описания) используются для иллюстрации альтернативной точки зрения, для отображения отдельных деталей, которые не поддерживаются явно синтаксисом IDEF0.

Событийная цепочка процессов (EPC, event-driven process chain) – тип диаграмм, используемых для моделирования, анализа и реорганизации бизнес-процессов (функционального моделирования).

Для информационной модели будет применена методология IDEF1X.

Методология IDEF1X подразделяется на уровни, соответствующие проектируемой модели данных системы. Каждый такой уровень соответствует определенной фазе проекта. Реализована в MSSQL.

Для поведенческих моделей будут применяться методологии:

* блок-схема – Basic Flowchart Shapes;
* EPC-диаграмма – EPC Diagram Shapes;
* BPmn-диаграмма – *Business Process Model and Notation.*

**Процесс (Basic Flowchart, простая блок-схема)** – нотация, представляющая собой простой вариант пошагового выполнения алгоритма. Используется на низшем уровне описания бизнес-модели.

Нотация EPC (Event-Driven Process Chain - событийная цепочка процессов) используется для описания процессов нижнего уровня. Диаграмма процесса в нотации EPC, представляет собой упорядоченную комбинацию событий и функций. Для каждой функции могут быть определены начальные и конечные события, участники, исполнители, материальные и документальные потоки, сопровождающие её, а также проведена декомпозиция на более низкие уровни. Декомпозиция может производиться в нотациях EPC или BPMN.

Функциональная модель системы учета работы учителя-логопеда по методологии IDEF0 представления на рисунке 2.1. Данная модель имеет 2 входных параметра: список детей, зачисленных на ЛП, выписка из протокола ПМПК (диагноз, рекомендации). 1 элемент управления: инструкции и указания ПМПК. 1 механизм: учитель-логопед. И 2 выхода – выписка из логопедического пункта и индивидуальная речевая карта.

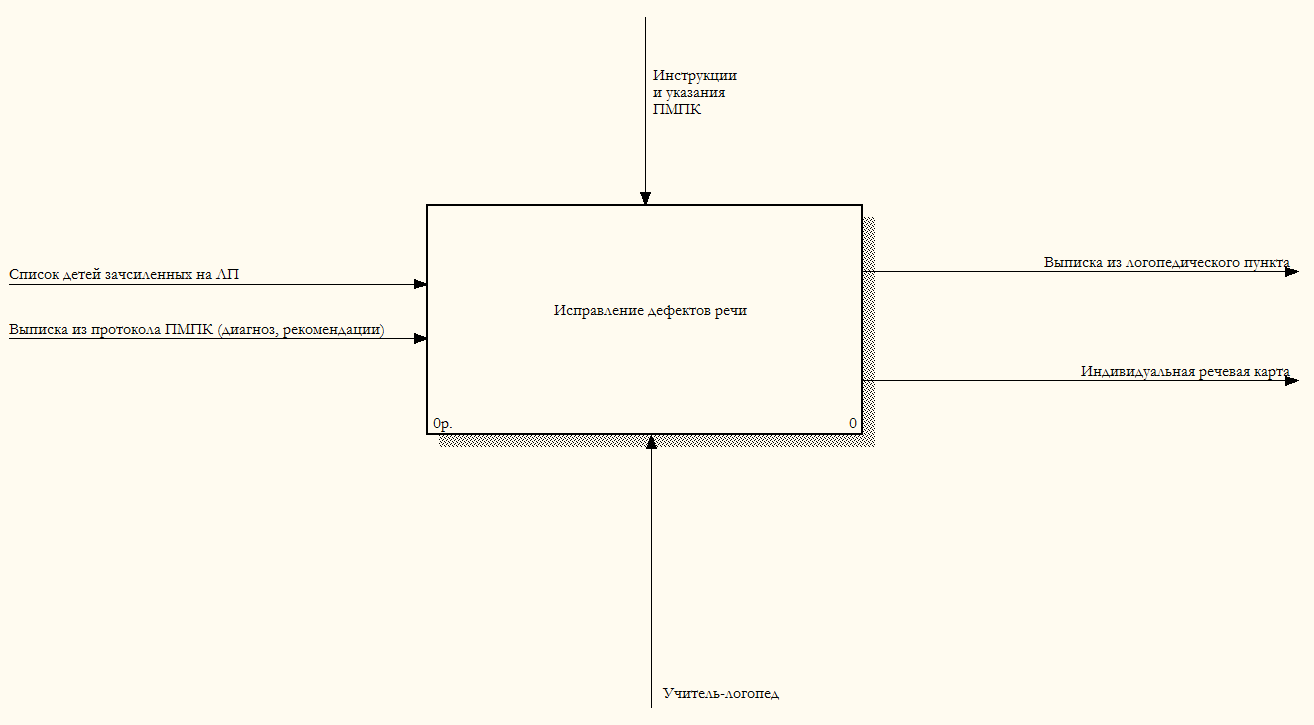


Рисунок 2.1 - Функциональная модель системы по методологии IDEF0

Диаграмма декомпозиции первого уровня. Диаграмма декомпозиции первого уровня системы учета работы учителя-логопеда представлена на рисунке 2.2. Данная диаграмма содержит в себе 5 работ: составление групп и расписания занятий; cоставление индивидуального плана работы; занятия с детьми; ведение и учет документации; итоговое заключение, диагноз. Каждый, из которых имеет свои входы, управления, механизмы и выходы.

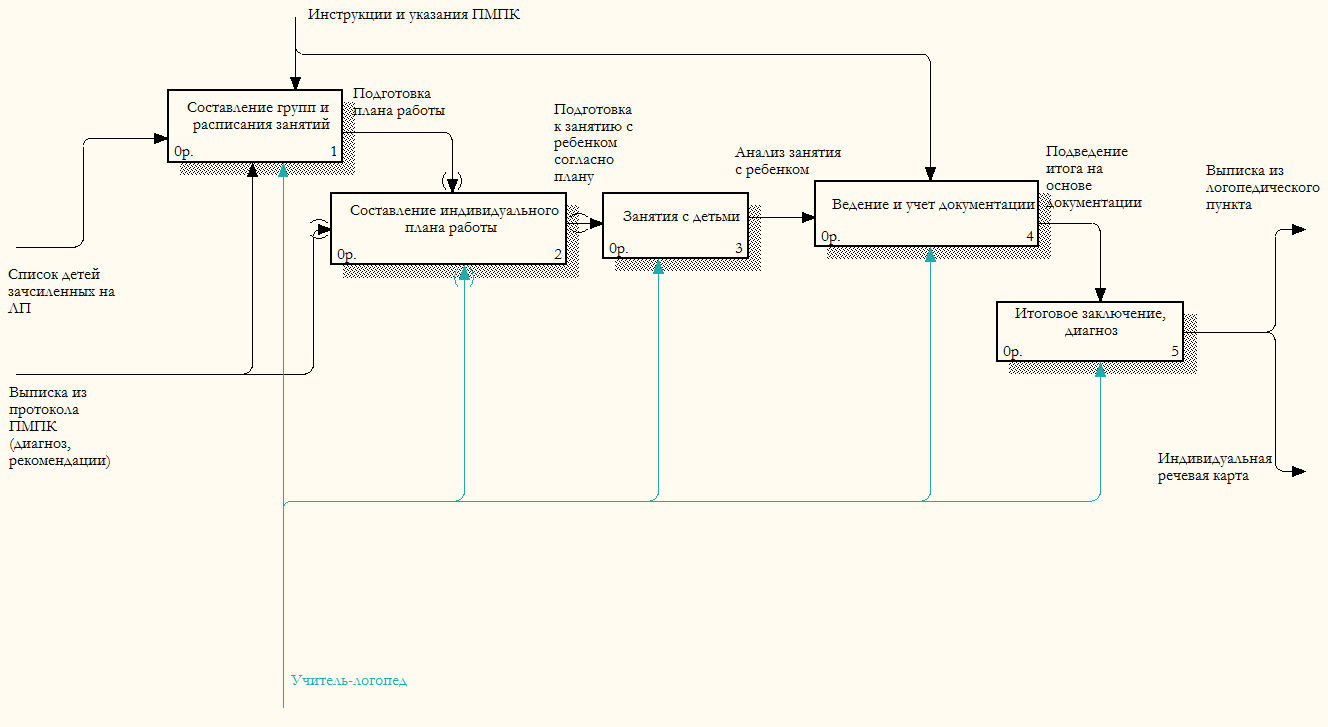


Рисунок 2.2 - Диаграмма декомпозиции первого уровня

Диаграмма содержит 5 работ, каждая из которых имеет входы, управления и механизмы.

Диаграмма декомпозиции второго уровня «Составление групп и расписания занятий» представления на рисунке 2.3. Данная диаграмма имеет два входа – список детей, зачисленных на ЛП и выписка из протокола ПМПК. 1 механизм: учитель-логопед. 1 элемент управления: инструкции и указания ПМПК. И один выход – подготовка плана работы.

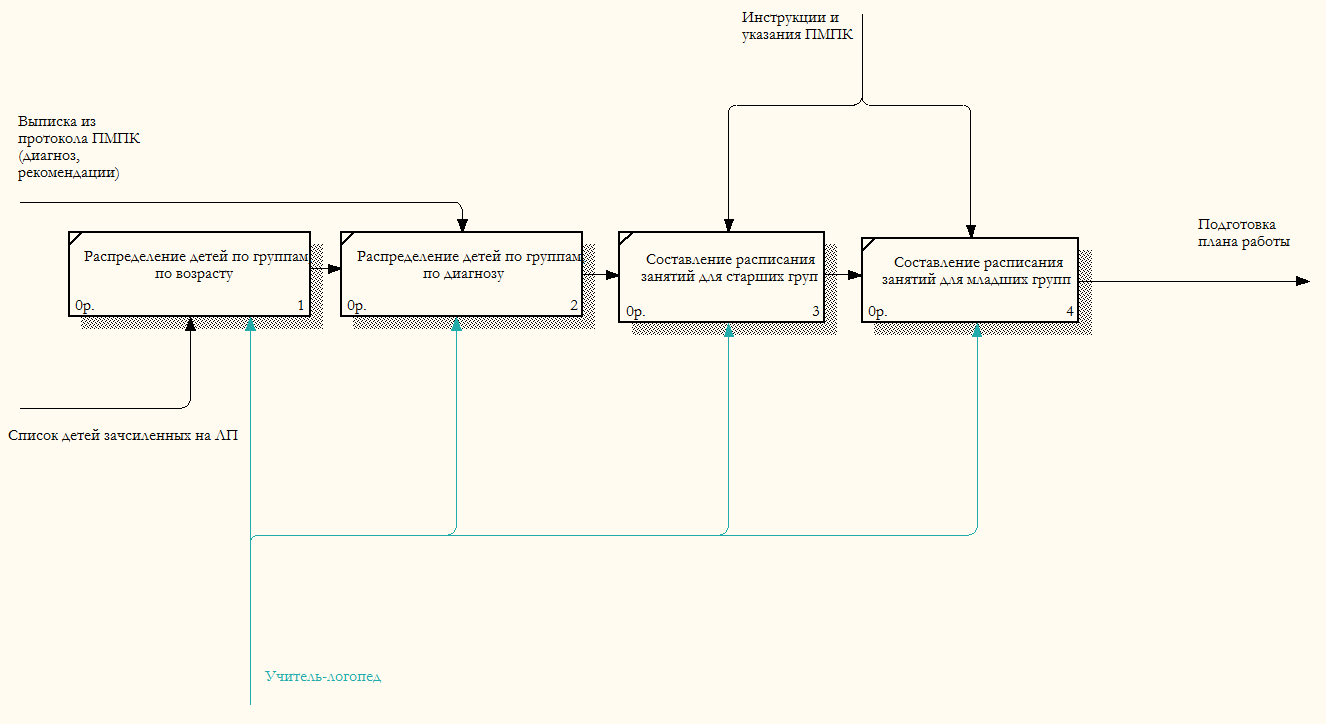


Рисунок 2.3 - Диаграмма декомпозиции второго уровня «Составление групп и расписания занятий»

Диаграмма имеет 2 входа, механизм, элемент управления и 1 выход.

Диаграмма декомпозиции второго уровня «Занятия с детьми». Диаграмма декомпозиции второго уровня «Занятия с детьми» представления на рисунке 2.4. Данная диаграмма имеет 1 вход: Подготовка к занятию с ребенком согласно плану. Один механизм: учитель-логопед. И один выход – Анализ занятия с ребенком.

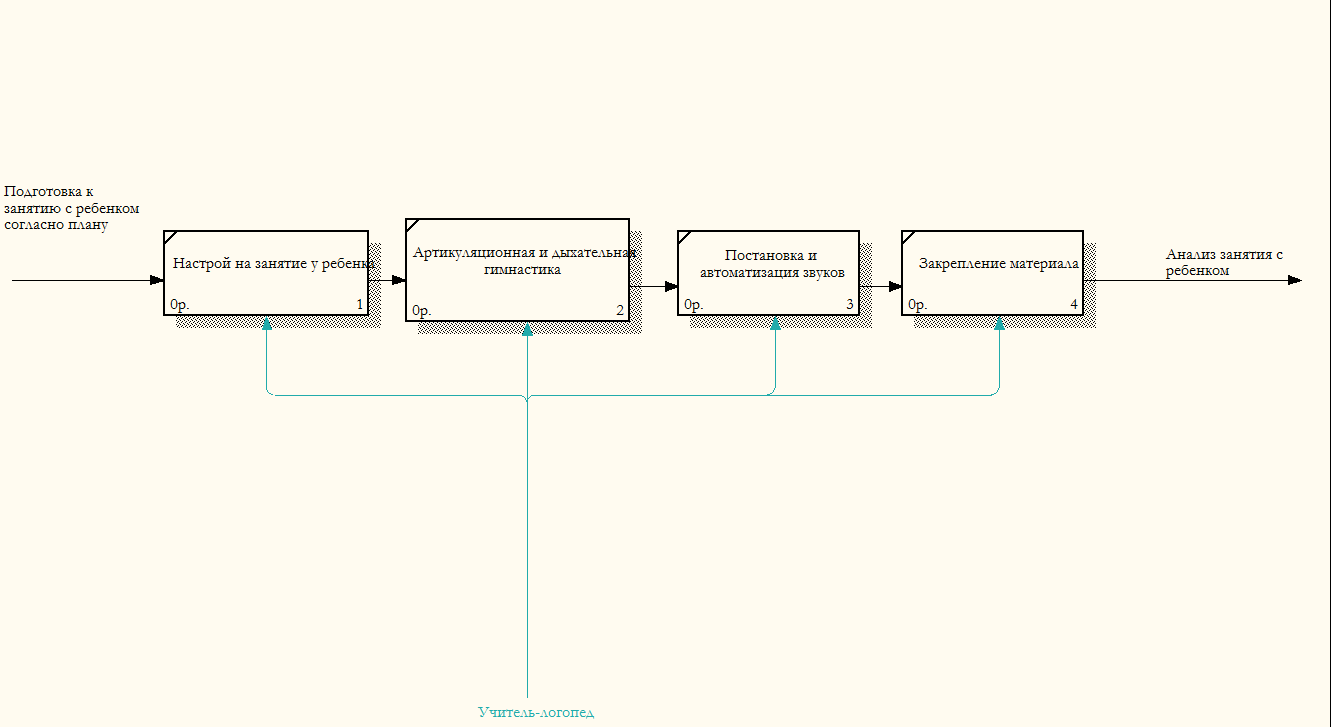


Рисунок 2.4 - Диаграмма декомпозиции второго уровня «Занятия с детьми»

Диаграмма декомпозиции второго уровня «Ведение и учет документации» представления на рисунке 2.5. Данная диаграмма имеет 1 вход: анализ занятия с ребенком. Один механизм: учитель-логопед. 1 элемент управления: инструкции и указания ПМПК. И один выход – подведение итога на основе документации.

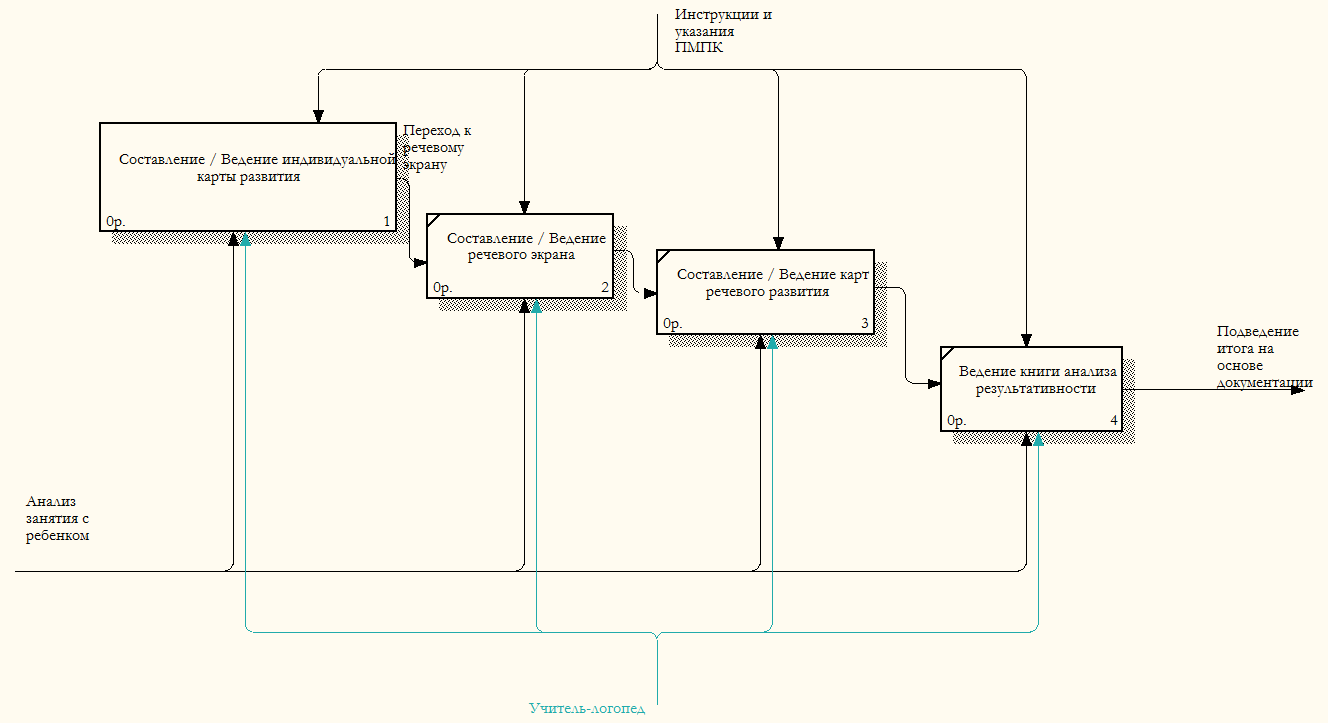


Рисунок 2.5 - Диаграмма декомпозиции второго уровня «Ведение и учет документации»

Диаграмма декомпозиции второго уровня «Итоговое заключение, диагноз».

Диаграмма декомпозиции второго уровня «Итоговое заключение, диагноз» представления на рисунке 2.6. Данная диаграмма имеет 1 вход: подведение итога на основе документации. 1 механизм: учитель-логопед. И два выхода – выписка из логопедического пункта и индивидуальная речевая карта.

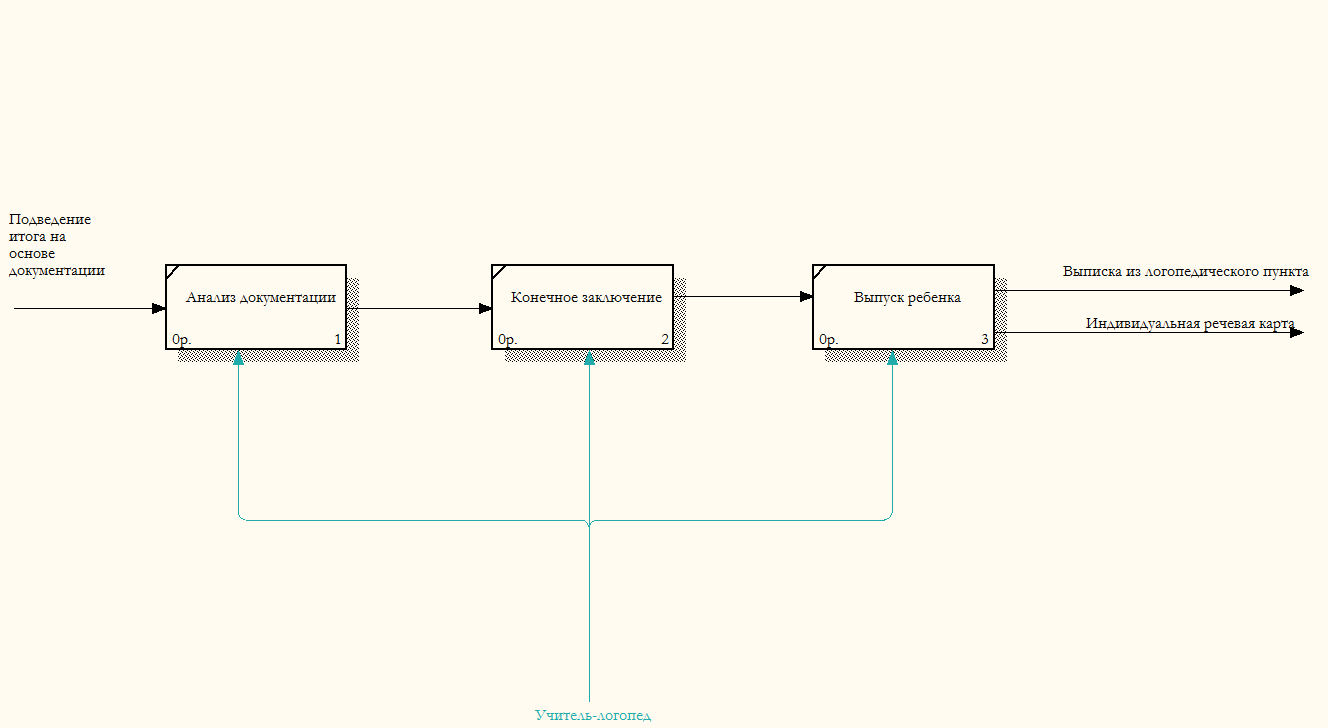


Рисунок 2.6 - Диаграмма декомпозиции второго уровня «Итоговое заключение, диагноз»

На диаграмме 1 вход, 1 механизм, и два выхода - выписка из логопедического пункта и индивидуальная речевая карта.

* + - 1. Функціональна модель системи за методологією IDEF3

«Составление / Ведение индивидуальной карты развития». Функциональная модель системы по методологии IDEF3 «Составление / Ведение индивидуальной карты развития» представлена на рисунке 2.7. Рассмотрим основные особенности этой диаграммы. После анализа занятия с ребенком, учитель-логопед может заполнить карту, если это было первое занятие в начале учебного года. Если же нет, тогда проявляется следующая ситуация. Нужно либо внести какие-то изменения, либо не вносить и перейти к следующей документации – Речевому экрану. Если необходимо все-таки внести правки, то можно отредактировать в первую очередь такие важные поля как звукопроизношение, слоговую структуру слова и моторику. И уже затем перейти к речевому экрану.

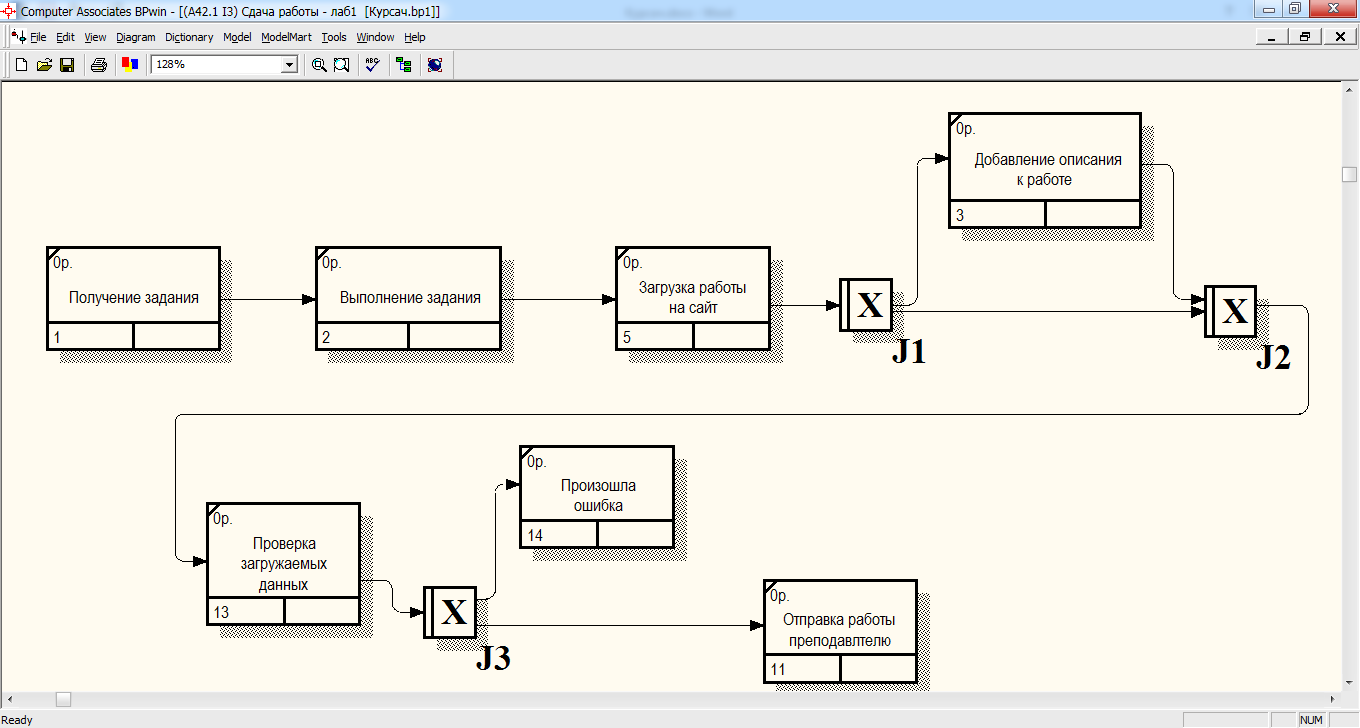


Рисунок 2.7 - Функциональная модель системы по методологии IDEF3 «Составление / Ведение индивидуальной карты развития»

На данном рисунке изображена функциональная модель составления и ведения индивидуальной карты развития.

* + - 1. Модель потоків даних

Модель потоков данных DFD «Составление индивидуального плана работы» представлена на рисунке 2.8. Центральным здесь является «Составление индивидуального плана на основе рекомендаций». На вход поступают «Выписка из протокола ПМПК» и «Подготовка плана работы». Также есть «Внесение корректив в существующий план». Выходом является «Подготовка к занятию с ребенком согласно плану»

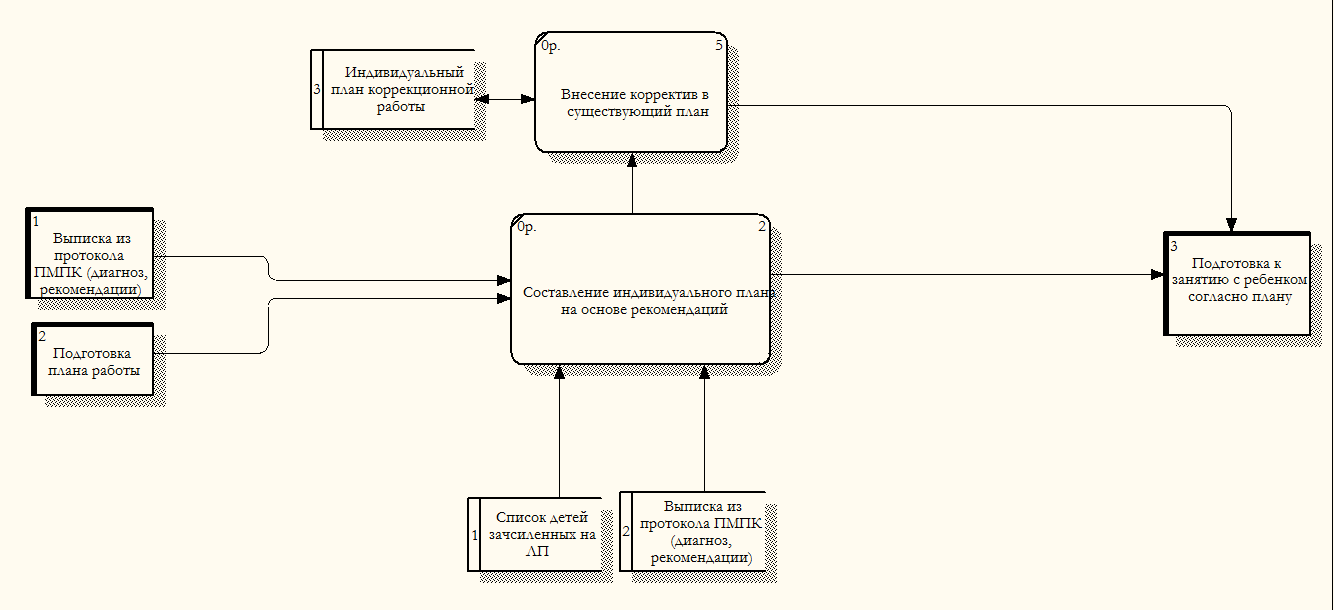


Рисунок 2.8 - Модель потоков данных DFD «Составление индивидуального плана работы»

На данном рисунке изображена модель потоков данных составления индивидуального плана работы по DFD.

* + - 1. Диаграмма дерева узлов

Диаграмма дерева узлов представлена на рисунке 2.9. Данная диаграмма показывает нам, какие работы и на каком уровне располагаются, в данном проект.

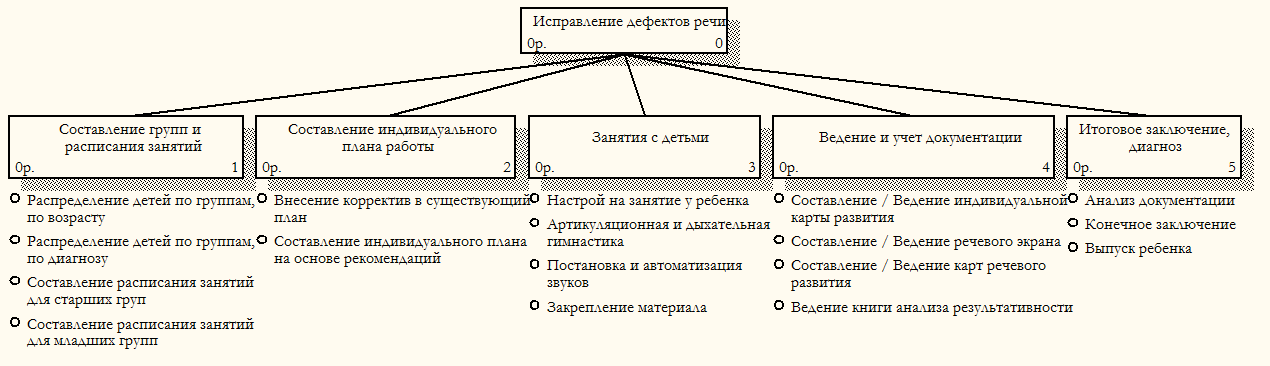


Рисунок 2.9 - Диаграмма дерева узлов

На рисунке представлено дерево узлов, уровни работы.

* + - 1. FEO-діаграми

FEO-диаграмма представлена на рисунке 2.10. Она показывает, как дочерние работы связаны между собой.

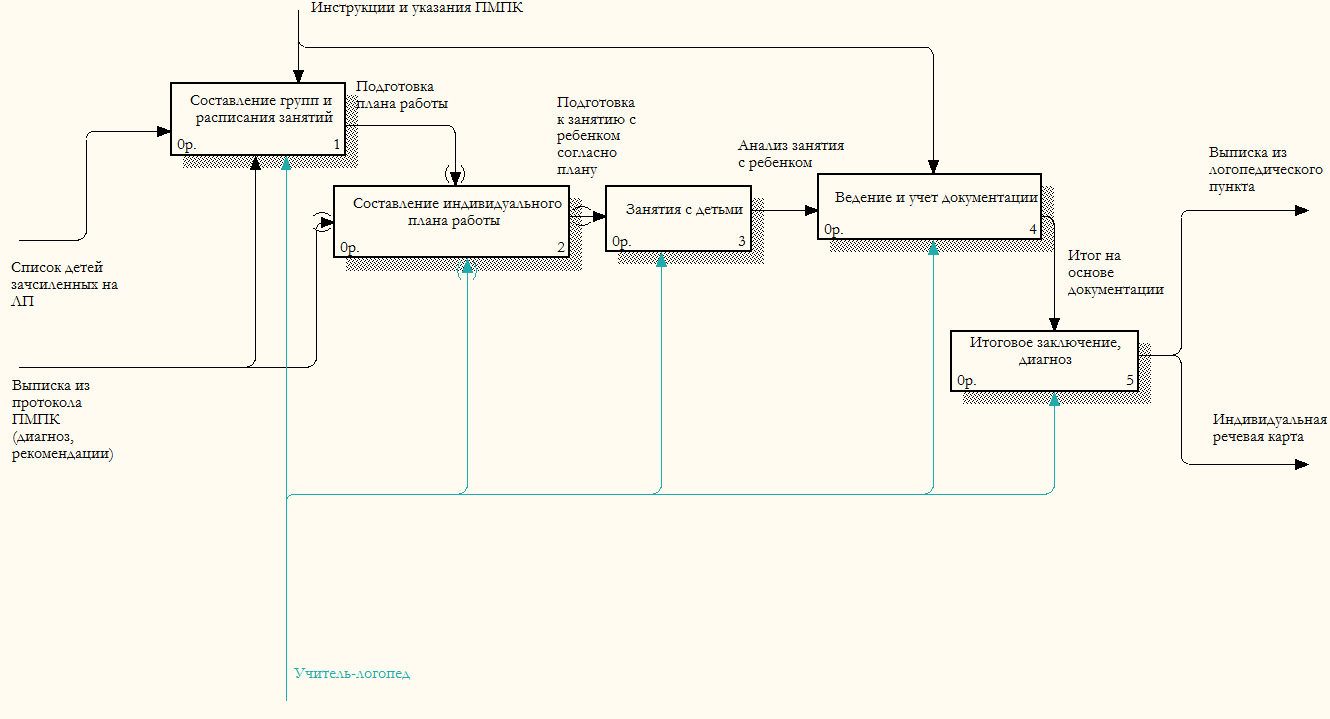


Рисунок 2.10 - FEO-диаграмма

Связь дочерних работ на FEO-диаграмме изображена на этом рисунке.

* + 1. Методы и средства объектно-ориентированного анализа и проектирования
       1. Диаграмма вариантов использования

В данной подсистеме существуют три пользователя, которые непосредственно работают с программой, либо с базой данных программы, а именно:

* учитель-логопед;
* заведующая психолого-медико-педагогической консультации;
* администратор.

Актёр «Учитель-логопед» это специалист, который занимается коррекцие йнарушений речи у детей, ведёт учёт всей документации (Карты речевого развития, планы работы, движение детей на ЛП, речевой экран, карты развития, картотеку игр, расписание занятий, список детей в ЛП, анализ результативности) и отправляет из программы результаты и отчёты заведующей ПМПК через факс.

Актёр «Заведующая ПМПК» это специалист, который координирует работу всей психолого-медико-педагогической консультации. Она следит за всеми документациями с разных логопедических пунктов через общую базу данных, в которой, собственно, и хранятся все выше перечисленные документы. Также она составляет график работы каждому учителю-логопеду в каждом логопедическом пункте.

Актёр «Администратор». Администратор занимается созданием и редактированием учетных записей.

На рисунке 2.12 изображена диаграмма прецедентов для проектируемой задачи.

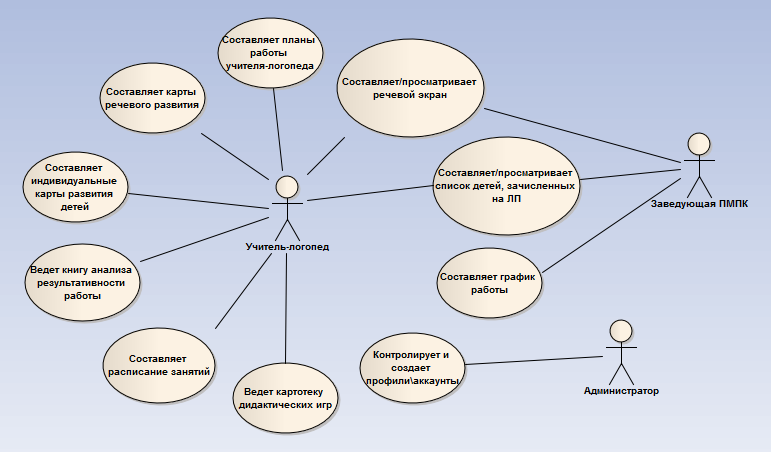


Рисунок 2.11 – Диаграмма вариантов использования

Пользователи, которые непосредственно работают в системе отображены на рисунке.

* + - 1. Описание варианта использования

Учитель-логопед будет взаимодействовать с такими прецендентами, как:

─ составляет планы работы;

─ составялет карты речевого развития;

─ составляет индивидуальные карты развития детей;

─ ведет книгу анализа результативности работы;

─ составляет расписание занятий;

─ составляет речевой экран;

─ составляет список детей зачисленных на ЛП.

Администратор будет взаимодействовать с такими прецендентами, как:

─ контроль и создание аккаунтов.

Заведующая ПМПК взаимодействует с прецендентами:

─ просматривает речевой экран;

─ просматривает список детей зачисленных на ЛП;

─ составляет график работы.

* + - 1. Диаграмма деятельности (activity diagram)

Разрабатываемая система характеризуется не только структурой составляющих ее элементов, но также и поведением (функциональностью). При моделировании поведения проектируемой системы возникает необходимость моделирования логической реализации выполняемых системой операций.

Диаграмма деятельности фокусируется на последовательности выполнения (потоке) и взаимосвязи действий (элементарных операций) в составе единого процесса, которые в совокупности приводят к получению желаемого результата.

В соответствии с вышеизложенными соображениями, диаграмма деятельности модели проектируемой информационной системы должна выглядеть, как это показано на рисунке 2.12

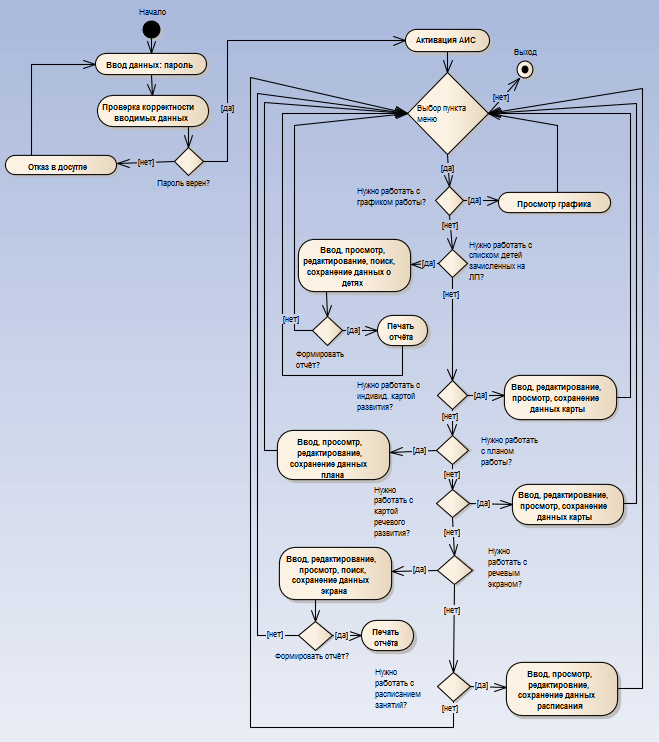


Рисунок 2.12 – Диаграмма деятельности АИС

Диаграмма деятельности модели проектируемой информационной системы.

* + - 1. Диаграмма последовательности (Sequence diagram)

Диаграмма последовательностей отражает взаимодействие объектов в динамике. В UML взаимодействие объектов понимается как обмен информацией между ними. При этом информация приобретает вид сообщений. Кроме того, что сообщение несет какую-то информацию, оно определенным образом также влияет на получателя. Как видим, в этом плане UML полностью соответствует основным принципам ООП, согласно которым информационное взаимодействие между объектами сводится к отправке и приему сообщений.

Диаграмма последовательностей относится к диаграммам взаимодействия UML, описывает поведенческие аспекты системы, но рассматривает взаимодействие объектов во времени. Другими словами, диаграмма последовательностей отражает временные особенности передачи и приема сообщений объектами.

Опытный читатель, возможно, скажет, что что-то подобное делает и диаграмма прецедентов. Использовать для уточнения диаграмм прецедентов, более детального описания логики сценариев использования. Это отличное средство документирования проекта с точки зрения сценариев использования! Диаграммы последовательностей обычно содержат объекты, которые взаимодействуют в рамках сценария, сообщений, они обмениваются, и которые возвращаются результаты, связанные с сообщениями. Впрочем, часто возвращаются результаты обозначают лишь в том случае, если это не очевидно из контекста.

На рисунке 2.13 - диаграмма последовательности взаимодействия.

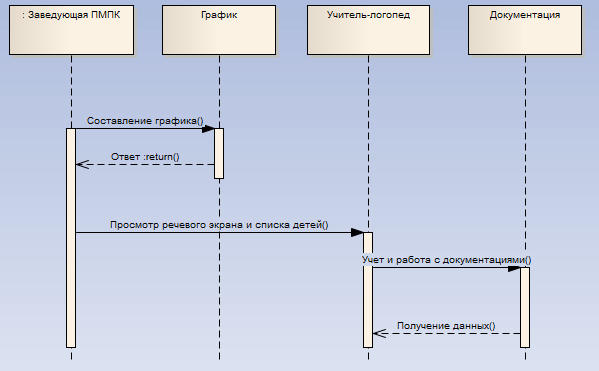


Рисунок 2.13 – Диаграмма последовательности

На рисунке изображена диаграмма последовательности для проектируемой системы.

* + - 1. Диаграмма классов

Диаграмма классов — диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. Входит в UML. На рисунке 2.14 представлена диаграмма классов.

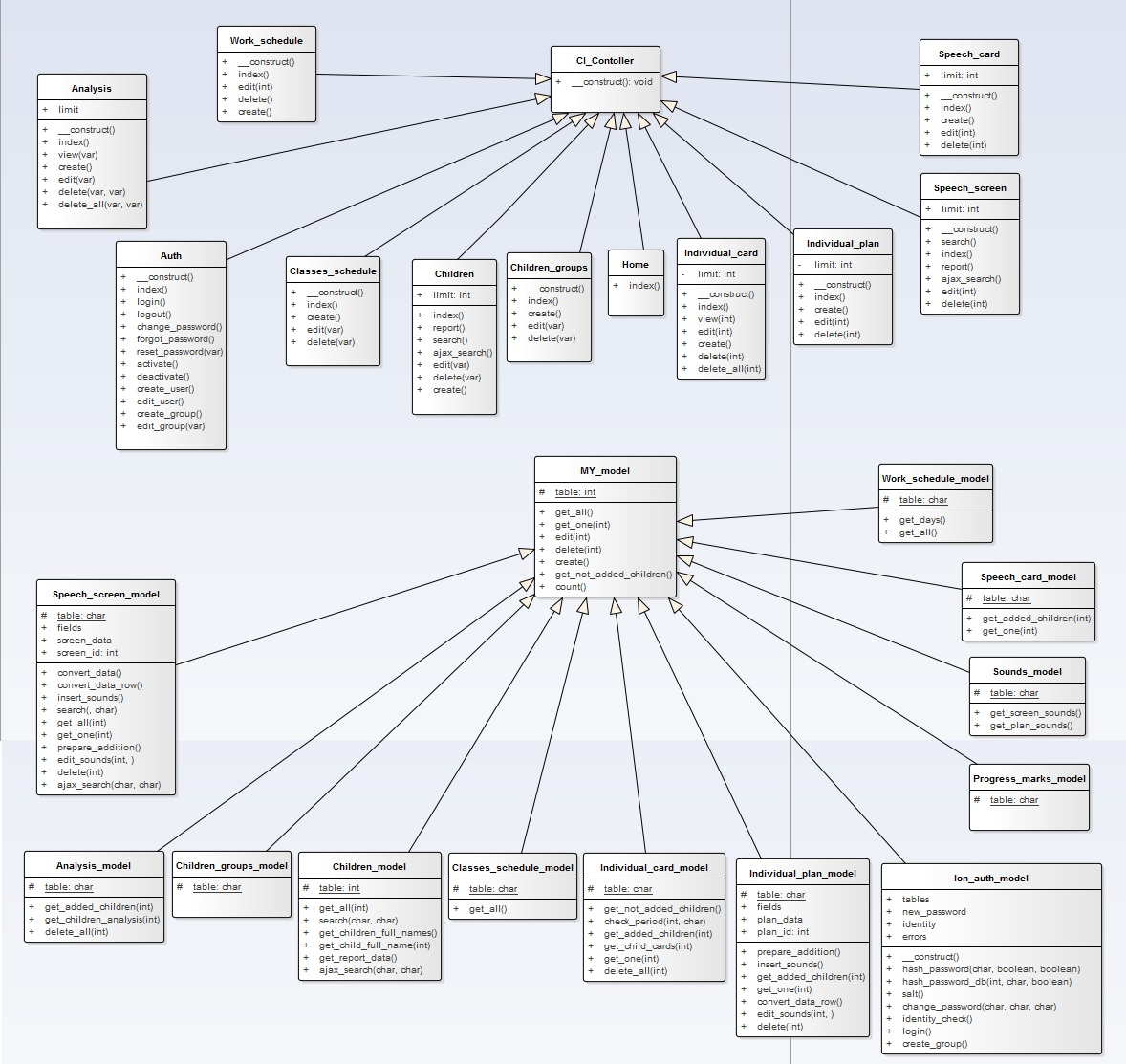


Рисунок 2.14 – Диаграмма классов

Диаграмма всех классов, методов и атрибутов проектируемой системы, а также их взаимосвязей.

* + - 1. Діаграма станів і переходів (state transition diagram)

Диаграмма состояний и переходов (state transition diagram) для одного из объэктов представлена на рисунке 2.15.

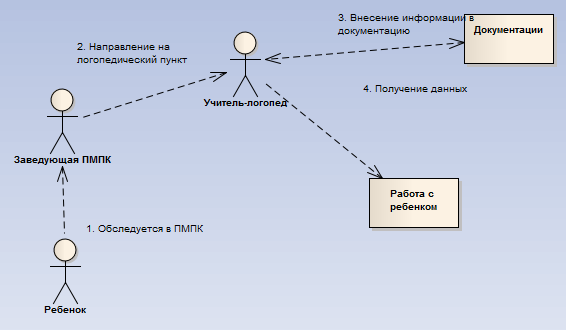


Рисунок 2.15 – Диаграмма состояний и переходов

На данной диаграмме изображены состояния и переходы в виде посследовательности

* 1. Інформаційне забезпечення

Исходные данные: выписка из протокола ПМПК, документ о речевом развитии ребенка, рекомендации ПМПК, замечания родителей.

* + 1. Концептуальная модель АИС

Концептуа́льная моде́ль — это модель, представленная множеством понятий и связей между ними, определяющих смысловую структуру рассматриваемой предметной области или её конкретного объекта.

Концептуальная модель — модель предметной области, состоящей из перечня взаимосвязанных понятий, используемых для описания этой области, вместе со свойствами и характеристиками, классификацией этих понятий, по типам, ситуациям, признакам в данной области и законов протекания процессов в ней. (Толковый словарь по искусственному интеллекту). Представлены таблицы с полями и их взаимосвязи. На рисунке 2.16 изображаена концептуальная модель АИС.

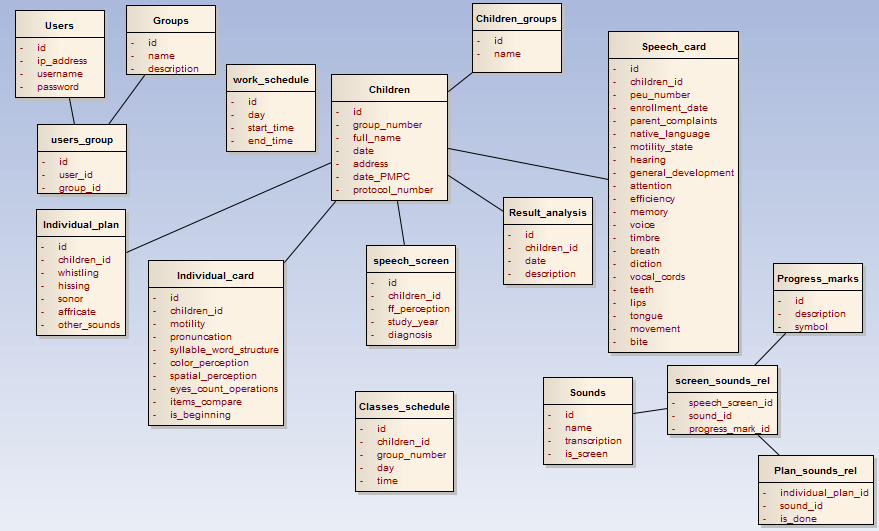


Рисунок 2.16 – Концептуальная модель

Модель предметной области, состоящей из перечня взаимосвязанных понятий.

* + 1. Даталогическая модель АИС

Под даталогической понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом даталогическая модель разрабатывается с учётом конкретной реализации СУБД, также с учётом специфики конкретной предметной области на основе её инфологической модели.

На рисунке 2.17 изобажена даталогическая модель.

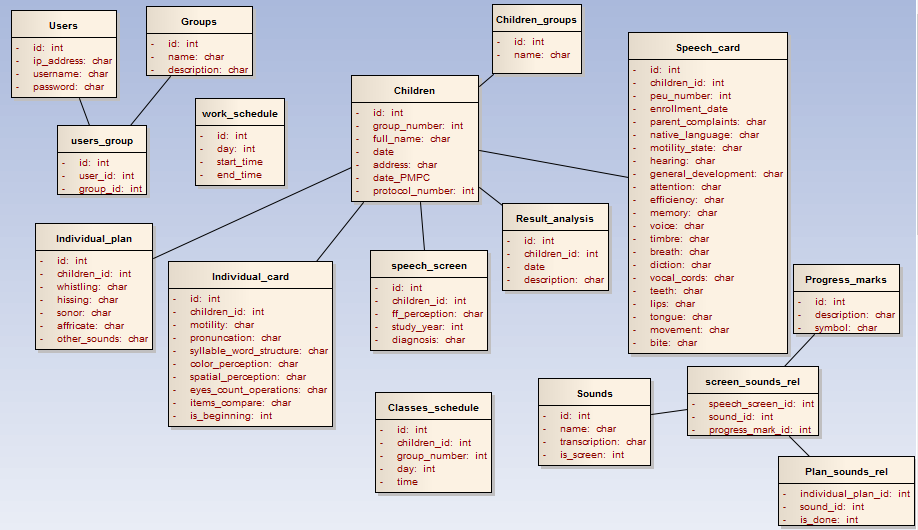


Рисунок 2.17 – Даталогическая модель

Даталогическая модель, таблицы, поля и их типы.

* + 1. Разработка алгоритмов решения функциональной задачи

Поведенческая модель системы Flowchart представлена на рисунке 2.18. Здесь изображена блок-схема составления/изменения индивидуального плана.

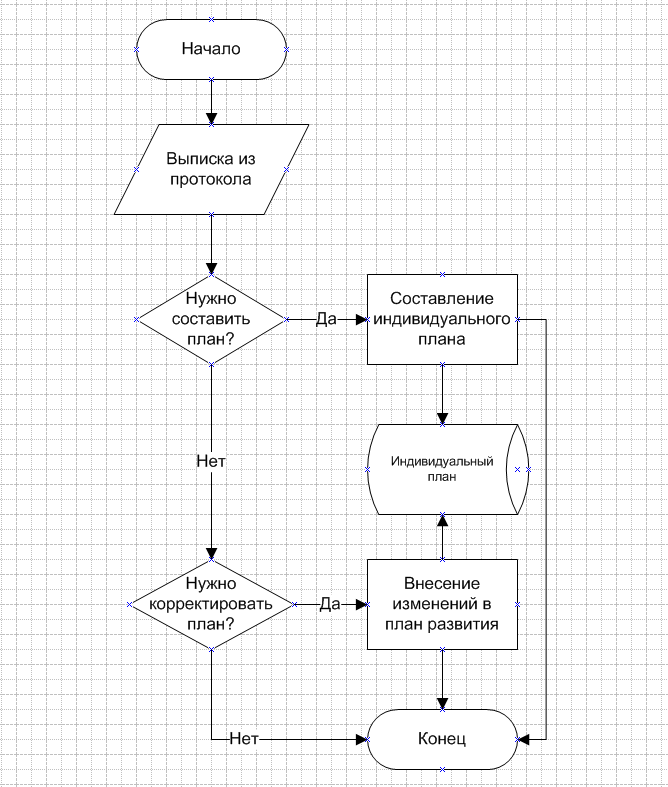


Рисунок 2.18 - Поведенческая модель системы Flowchart

Блок-схема составления/изменения индивидуального плана

* 1. Программное обеспечение

Обоснование выбора системы управления базами данных

Для разрабатываемой мною задачи используется клиент серверная архитектура, потому что с данной базой данных будут работать заведующая психолого-медико-педагогической консультации и учителя-логопеды данной консультации в разных логопедических пунктах.

Так как архитектура программного обеспечения для поставленной задачи будет клиент серверной, то в качестве СУБД мною была выбрана реляционная СУБД MySQL, поскольку эта СУБД быстрая, бесплатная и надёжная.

MySQL является собственностью компании Oracle Corporation, получившей её вместе с поглощённой Sun Microsystems, осуществляющей разработку и поддержку приложения. Распространяется под GNU General Public License или под собственной коммерческой лицензией. Помимо этого разработчики создают функциональность по заказу лицензионных пользователей, именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Несмотря на то что MySQL постоянно совершенствуется, он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей доступности, скорости и безопасности MySQL очень хорошо подходит для доступа к базам данных по Internet.

Технические возможности СУБД MySQL. MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API).

Cтруктура MySQL трехуровневая: базы данных — таблицы — записи. Базы данных и таблицы MySQL физически представляются файлами с расширениями frm, MYD, MYI. Логически - таблица представляет собой совокупность записей. А записи - это совокупность полей разного типа. Имя базы данных MySQL уникально в пределах системы, а таблицы - в пределах базы данных, поля - в пределах таблицы. Один сервер MySQL может поддерживать сразу несколько баз данных, доступ к которым может разграничиваться логином и паролем. Зная эти логин и пароль, можно работать с конкретной базой данных. Например, можно создать или удалить в ней таблицу, добавить записи и т. д. Обычно имя-идентификатор и пароль назначаются хостинг провайдерами, которые и обеспечивают поддержку MySQL для своих пользователей.

Обоснование выбора языка программирования.

Главным фактором языка РНР является практичность. РНР должен предоставить программисту средства для быстрого и эффективного решения поставленных задач. Практический характер РНР обусловлен пятью важными характеристиками:

– традиционностью;

– простотой;

– эффективностью;

– безопасностью;

– гибкостью.

Существует еще одна «характеристика», которая делает РНР особенно привлекательным: он распространяется бесплатно! Причем, с открытыми исходными кодами (Open Source).

Традиционность. Язык РНР будет казаться знакомым программистам, работающим в разных областях. Многие конструкции языка позаимствованы из Си, Perl.

Код РНР очень похож на тот, который встречается в типичных программах на С или Pascal. Это заметно снижает начальные усилия при изучении РНР. PHP — язык, сочетающий достоинства Perl и Си и специально нацеленный на работу в Интернете, язык с универсальным (правда, за некоторыми оговорками) и ясным синтаксисом.

И хотя PHP является довольно молодым языком, он обрел такую популярность среди web-программистов, что на данный момент является чуть ли не самым популярным языком для создания web-приложений (скриптов).

Простота. Сценарий РНР может состоять из 10 000 строк или из одной строки — все зависит от специфики вашей задачи. Вам не придется подгружать библиотеки, указывать специальные параметры компиляции или что-нибудь в этом роде. Если код имеет правильный синтаксис, он исполняется в точности так, как указал программист.

PHP — язык, который может быть встроен непосредственно в html -код страниц, которые, в свою очередь будут корректно обрабатываться PHP -интерпретатором. Мы можем использовать PHP для написания CGI-сценариев и избавиться от множества неудобных операторов вывода текста. Мы можем привлекать PHP для формирования HTML-документов, избавившись от множества вызовов внешних сценариев.

Большое разнообразие функций PHP избавят вас от написания многострочных пользовательских функций на C или Pascal .

Эффективность. Эффективность является исключительно важным фактором при программировании для многопользовательских сред, к числу которых относится и web .

Очень важное преимущество PHP заключается в его «движке». «Движок» PHP не является ни компилятором, ни интерпретатором. Он является транслирующим интерпретатором. Такое устройство «движка» PHP позволяет обрабатывать сценарии с достаточно высокой скоростью [30].

По некоторым оценкам, большинство PHP-сценариев (особенно не очень больших размеров) обрабатываются быстрее аналогичных им программ, написанных на Perl. Однако, чтобы не делали разработчики PHP, откомпилированные исполняемые файлы будут работать значительно быстрее – в десятки, а иногда и в сотни раз. Но производительность PHP вполне достаточна для создания вполне серьезных web-приложений. Подробно об устройстве и характеристиках «движка» PHP можно ознакомиться в источнике [25, 26].

Безопасность. РНР предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные средства безопасности, которые условно делятся на две категории: средства системного уровня и средства уровня приложения.

Средства безопасности системного уровня. В РНР реализованы механизмы безопасности, находящиеся под управлением администраторов; при правильной настройке РНР это обеспечивает максимальную свободу действий и безопасность. РНР может работать в так называемом безопасном режиме (safe mode), который ограничивает возможности применения РНР пользователями по ряду важных показателей. Например, можно ограничить максимальное время выполнения и использование памяти (неконтролируемый расход памяти отрицательно влияет на быстродействие сервера). По аналогии с cgi-bin администратор также может устанавливать ограничения на каталоги, в которых пользователь может просматривать и исполнять сценарии РНР, а также использовать сценарии РНР для просмотра конфиденциальной информации на сервере (например, файла passwd).

Средства безопасности уровня приложения. В стандартный набор функций РНР входит ряд надежных механизмов шифрования. РНР также совместим с многими приложениями независимых фирм, что позволяет легко интегрировать его с защищенными технологиями электронной коммерции (e-commerce). Другое преимущество заключается в том, что исходный текст сценариев РНР нельзя просмотреть в браузере, поскольку сценарий компилируется до его отправки по запросу пользователя. Реализация РНР на стороне сервера предотвращает похищение нетривиальных сценариев пользователями, знаний которых хватает хотя бы для выполнения команды View Source.

Гибкость. Поскольку РНР является встраиваемым (embedded) языком, он отличается исключительной гибкостью по отношению к потребностям разработчика. Хотя РНР обычно рекомендуется использовать в сочетании с HTML, он с таким же успехом интегрируется и в JavaScript, WML, XML и другие языки. Кроме того, хорошо структурированные приложения РНР легко расширяются по мере необходимости (впрочем, это относится ко всем основным языкам программирования).

Нет проблем и с зависимостью от браузеров, поскольку перед отправкой клиенту сценарии РНР полностью компилируются на стороне сервера. В сущности, сценарии РНР могут передаваться любым устройствам с браузерами, включая сотовые телефоны, электронные записные книжки, пейджеры и портативные компьютеры, не говоря уже о традиционных ПК. Программисты, занимающиеся вспомогательными утилитами, могут запускать РНР в режиме командной строки.

Поскольку РНР не содержит кода, ориентированного на конкретный web-сервер, пользователи не ограничиваются определенными серверами (возможно, незнакомыми для них). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold и Zeus — РНР работает на всех перечисленных серверах. Поскольку эти серверы работают на разных платформах, РНР в целом является платформенно-независимым языком и существует на таких платформах, как UNIX, Solaris, FreeBSD и Windows 95/98/NT/2000/XP/2003.

Наконец, средства РНР позволяют программисту работать с внешними компонентами, такими как Enterprise Java Beans или СОМ-объекты Win32. Благодаря этим новым возможностям РНР занимает достойное место среди современных технологий и обеспечивает масштабирование проектов до необходимых пределов.

Бесплатное распространение. Стратегия Open Source, и распространение исходных текстов программ в массах, оказало несомненно благотворное влияние на многие проекты, в первую очередь — Linux, хотя и успех проекта Apache сильно подкрепил позиции сторонников Open Source. Сказанное относится и к истории создания РНР, поскольку поддержка пользователей со всего мира оказалась очень важным фактором в развитии проекта РНР.

Принятие стратегии Open Source и бесплатное распространение исходных текстов РНР оказало неоценимую услугу пользователям. Вдобавок, отзывчивое сообщество пользователей РНР является своего рода «коллективной службой поддержки», и в популярных электронных конференциях можно найти ответы даже на самые сложные вопросы.

* + 1. Схема взаимодействия программных модулей

На рисунке 2.19 изображена схема взаимодействий модулей

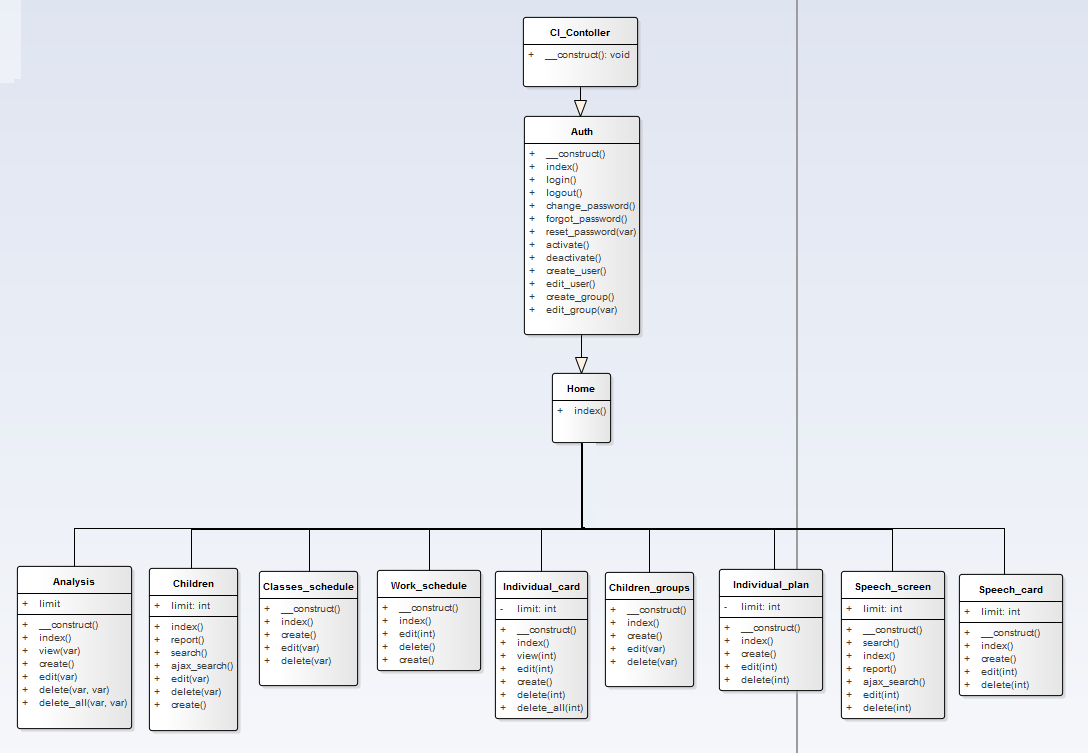


Рисунок 2.19 – Схема взаимодействия модулей

На диаграмме показано как взаимодейсвуют программные модули в ПО

* + 1. Архитектура системы

В основе проектирования АС лежит клиент-серверная архитектура.

Клиент-сервер представляет собой вычислительную или сетевую архитектуру, в которой задания или сетевая нагрузка распределены между поставщиками услуг, которые называются серверами, и заказчиками услуг, называемых клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением.

Данная архитектура имеет ряд преимуществ:

─ делает возможным, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети. Это позволяет упростить обслуживание вычислительной системы. В частности, замена, ремонт, модернизация или перемещение сервера, не затрагивают клиентов.

─ все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешить доступ к данным только клиентам с соответствующими правами доступа.

─ позволяет объединить различные клиенты. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами.

Наряду с достоинствами, существуют и недостатки архитектуры:

─ неработоспособность сервера может сделать неработоспособной всю вычислительную сеть.

─ поддержка работы данной системы требует отдельного специалиста - системного администратора.

─ высокая стоимость оборудования.

* + - 1. Диаграмма компонентов

Диаграмма компонентов и развертывания сделана вместе и сформирована в одну единственную диаграмму.

* + - 1. Диаграмма развертывания

Элементы логического представления (Классы и ассоциации) не существует физически, а лишь отражают понимание структуры системы или аспекты его поведения. В UML для этих целей есть два вида диаграмм: диаграмма развертывания и диаграмма компонентов.

Хотя диаграммы развертывания и диаграммы компонентов можно изображать отдельно, также допускается помещать диаграмму компонентов на диаграмму развертывания. Это целесообразно делать, чтобы показать какие компоненты выполняются и на каких узлах.

Диаграмма компонентов показывает зависимости и взаимодействия между компонентами программного обеспечения. Она изображает физическую архитектуру компьютера базовой системы. Диаграмма компонентов разрабатывается для визуализации общей структуры исходного кода и спецификации сборки исполняемого кода системы. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости. Диаграмма компонентов позволяет определить архитектуру разрабатываемой, установив зависимости между программными компонентами.

Компонент - это контейнер пакетов, участвующих в работе системы. Рабочие компоненты представлены пакетами выходного и исполняемого кода. Для взаимодействия компонентов системы компоненты имеют порты (для организации взаимодействия распределенных компонентов) и реализуют некоторый набор интерфейсов.

Диаграмма компонентов (развертывание) модели проектируемой системы выглядит, как показано на рисунке 2.20.



Рисунок 2.20 - Диаграмма компонентов (развертывание) системы

Диаграммы развертывания - это один из двух видов диаграмм, используемых при моделировании физических аспектов объектно - ориентированной системы. Такая диаграмма показывает конфигурацию узлов, где проводится обработка информации, и то, какие компоненты размещены на каждом узле.

Диаграммы развертывания используются для моделирования статического вида системы с точки зрения развертывания. В основном под этим понимается моделирование топологии аппаратных средств, на которых выполняется система.

Диаграммы развертывания важны не только для визуализации, спецификации и документирования встроенных, клиент - серверных и распределенных систем, но и для управления исполнительными системами с использованием прямого и обратного проектирования.

* 1. Организационное обеспечение
     1. Инструкция пользователя

АИС учета работы учителя-логопеда предназначена для автоматизации учета документации и комфортной работы.

Приложение является сайтом, и чтобы начать пользоваться им, необходимо перейти по адресу веб-сайта.

После перехода на сайт, появится форма входа в учетную запись. На рисунке 2.21 изображена форма авторизации.

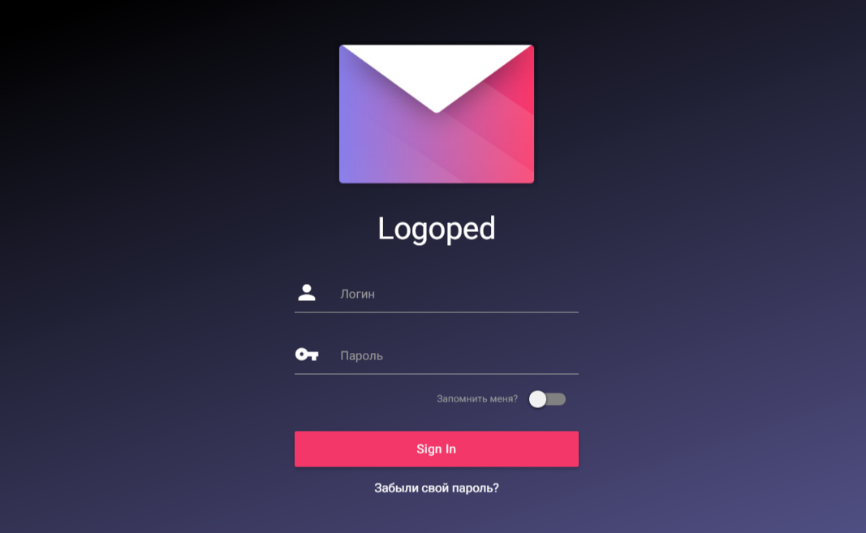


Рисунок 2.21 – Окно авторизации

В этом окне необходимо ввести логин и пароль от учетной записи. Также можно отметить пункт «Запомнить меня».

Существуют 3 типа пользователей: учитель-логопед, заведующая ПМПК, и администратор. Разберем основного пользователя – учителя-логопеда. На рисунке 2.22 изображена главная страница сайта после авторизации.

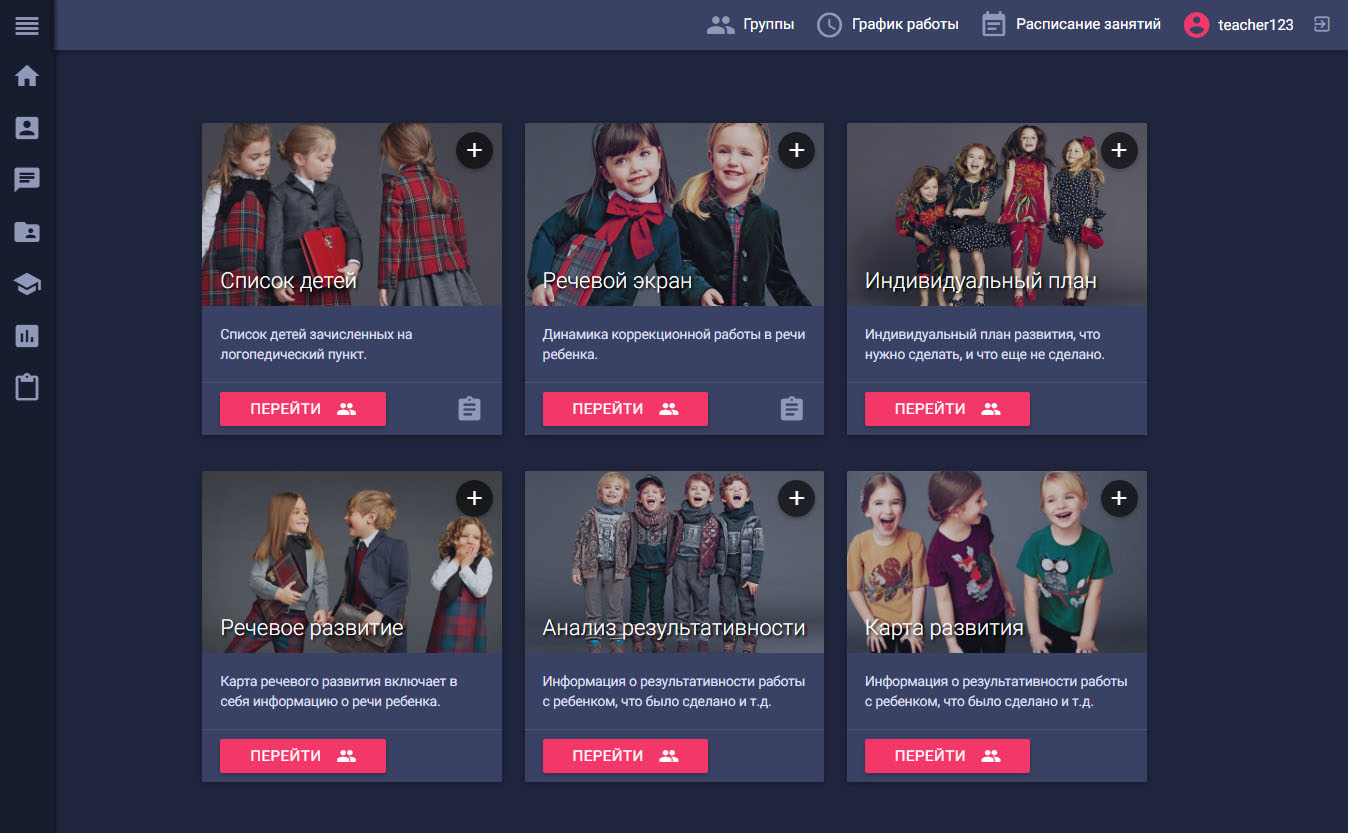


Рисунок 2.22 – Главная страница

На главной странице находится меню, состоящее из элементов навигации. Навигация доступна на любой странице сайта. В ней есть следующие элементы: элемент меню «Группы», которые откроют список групп, а также элементы меню «График работы», «Расписание занятий», индикатор аккаунта, и кнопка «Выйти», которая позволяет выйти из текущего аккаунта.

В основном разделе страницы, видны карточки, которые представляют собой одну из документаций, которые были автоматизированы. На каждой из карточек, присутствуют кнопки для перехода, и кнопки добавления. А также, для документаций «Список детей» и «Речевой экран» присутствуют кнопки для загрузки отчета.

Также, слева находится навигационная панель, с ссылками на каждую документацию. Если нажать на самую верхнюю кнопку в панели, то она раскроется в более детальный вид. Данное событие можно посмотреть на рисунке 2.23

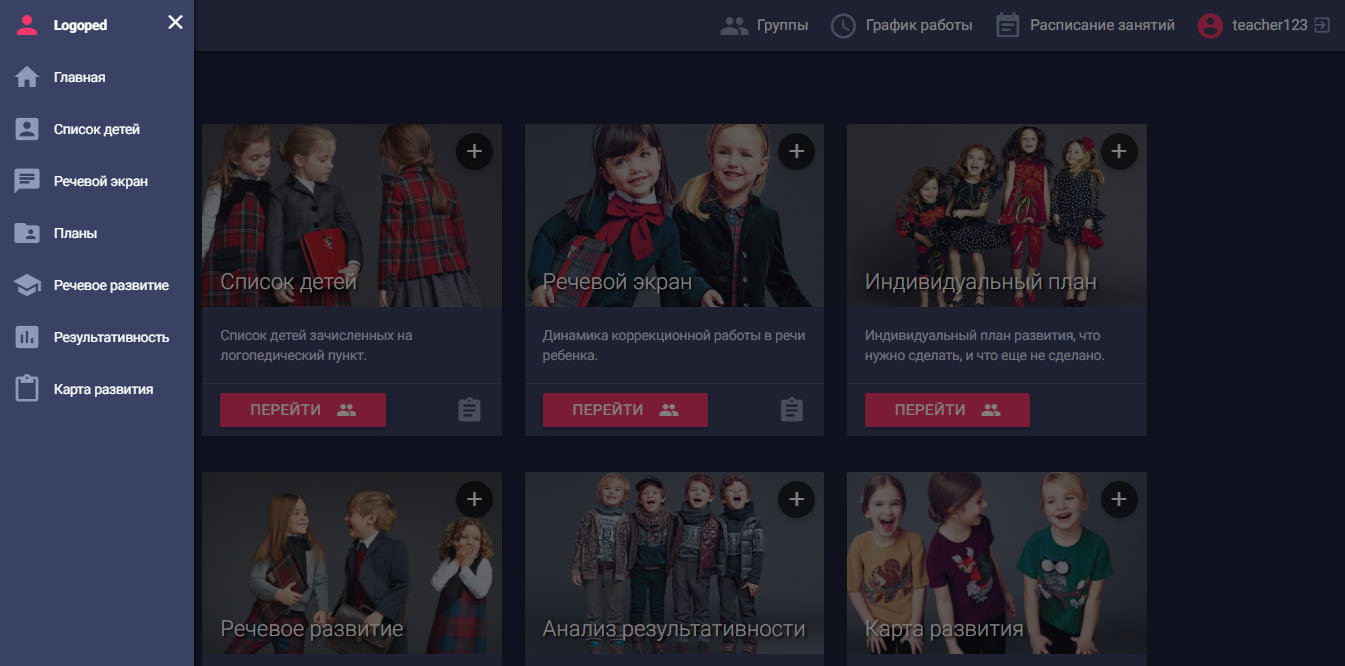


Рисунок 2.23 – Раскрытая панель навигации

Перейдя по первой документации «Список детей», откроется страница со списком детей зачисленных на ЛП. На рисунке 2.24 изображен список детей.

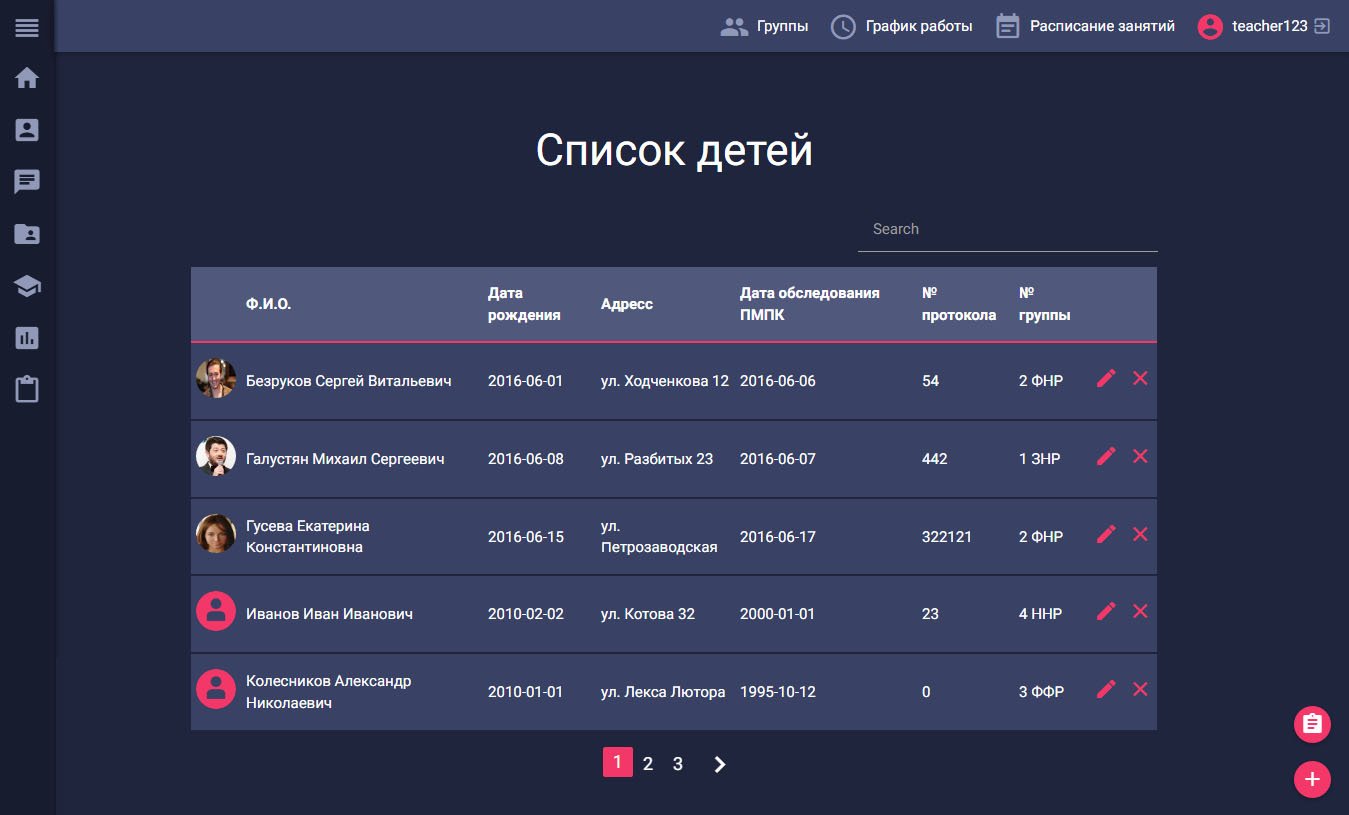


Рисунок 2.24 – Список детей, зачисленных на ЛП

В данной документации, отображены дети и информация о них. Кнопки редактирования и удаления записей. А также кнопка добавления нового ребенка в нижней правой части угла страницы, и кнопка загрузки отчета которая находится выше кнопки добавления. Также, на странице находится поле для поиска детей по фамилии. Нажав на кнопку добавления, откроется страница добавления нового ребенка (Рисунок 2.25).

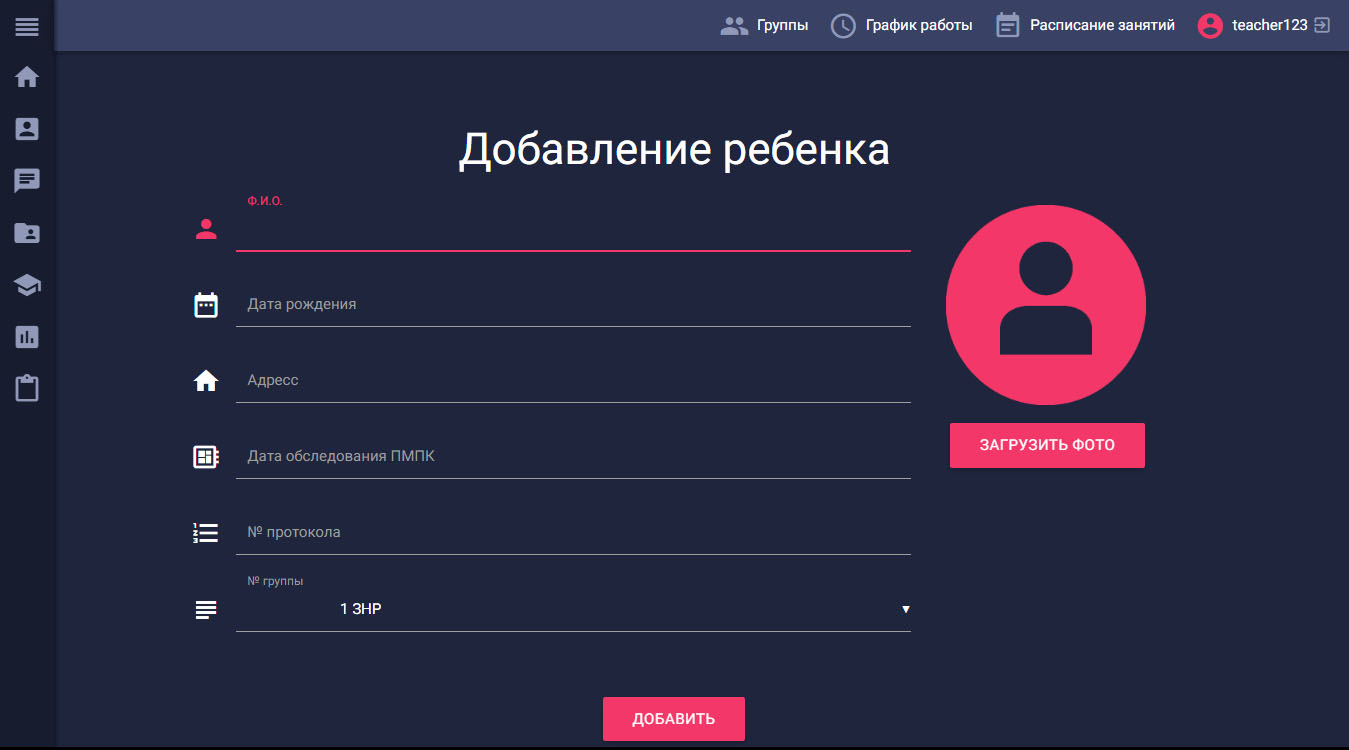


Рисунок 2.25 – Добавление ребенка

В этой странице указываются данные ребенка, и, если есть электронное фото, то есть возможность загрузить фото.

Также, для удобства был добавлен модуль добавления даты, он показан на рисунке 2.26.

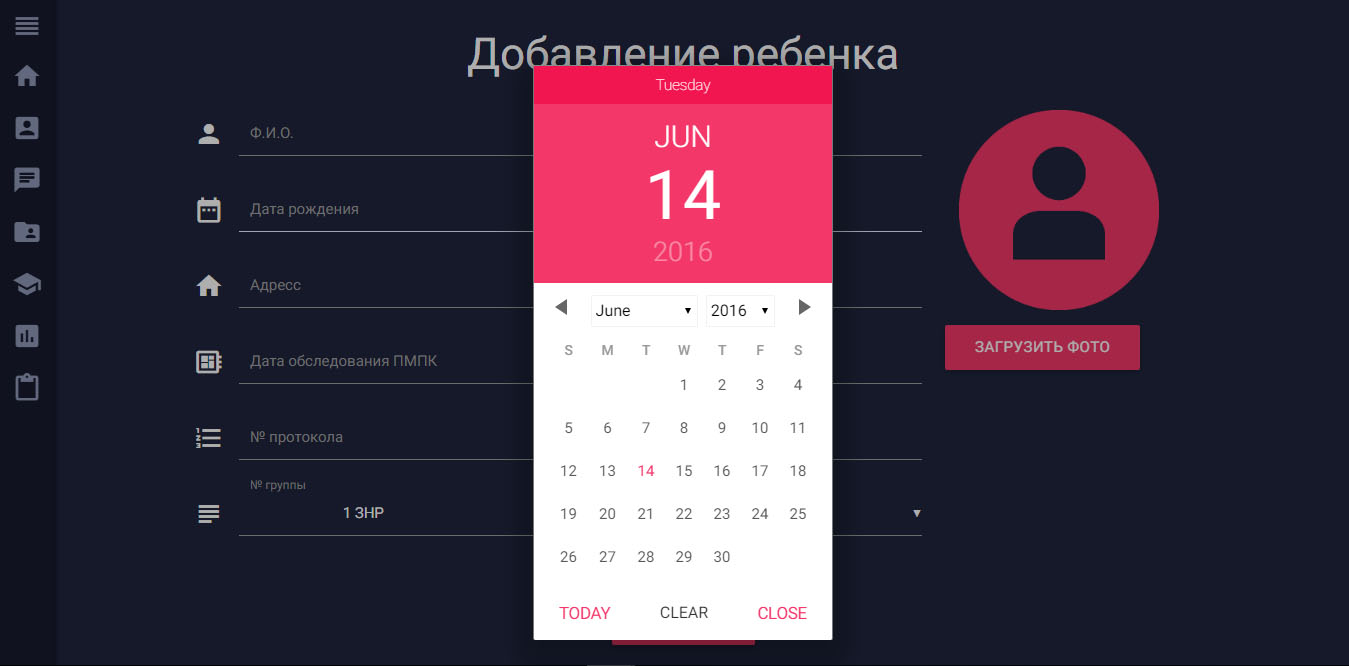


Рисунок 2.26 – Окно добавления даты

При нажатии на кнопку отчета, браузер предложит сохранить отчет в формате xls. Сформированный отчет в MS Excel показан на рисунке 2.27

.

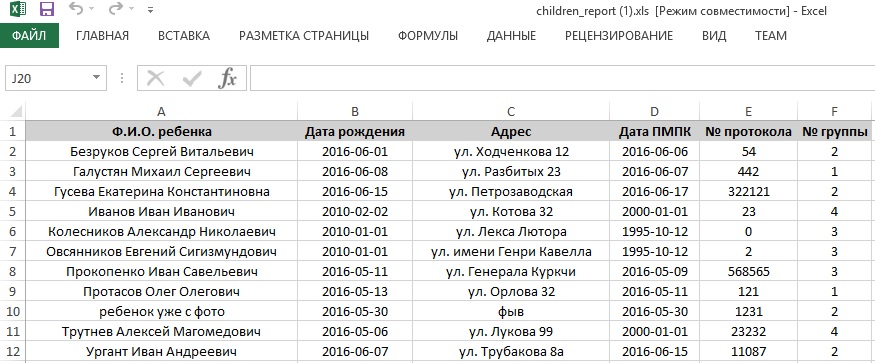


Рисунок 2.27 – Сформированный приложением отчет

На странице находится поле для поиска. Поиск работает как при нажатии клавишы Enter, так и просто при введении словосочетания. Как выглядит моментальный поиск, можно увидеть на рисунке 2.28.

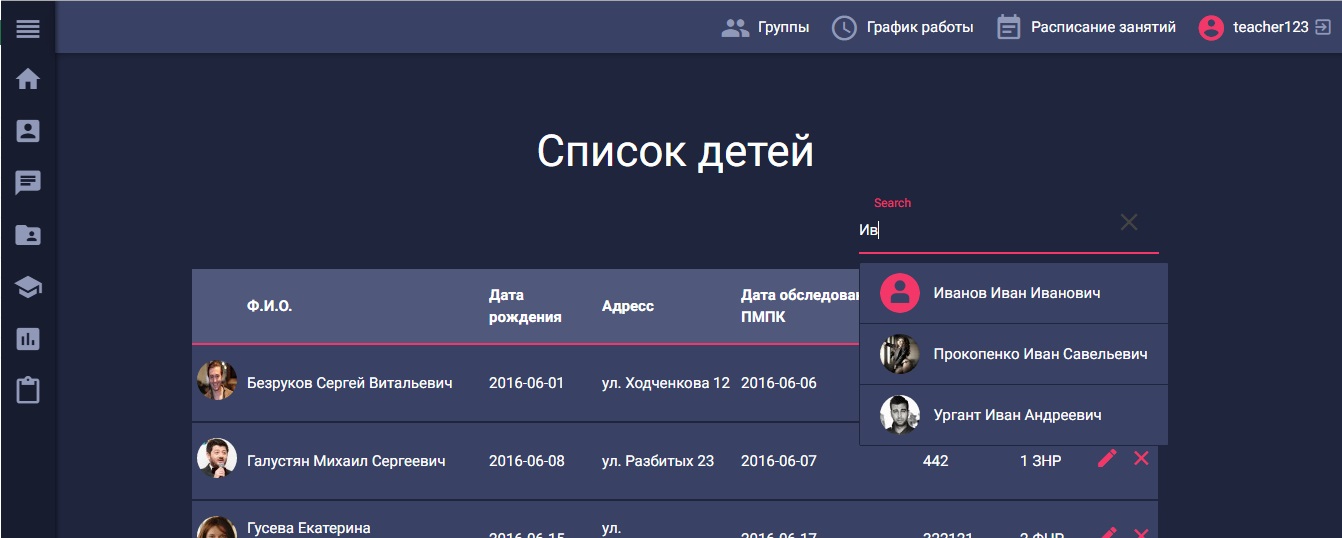


Рисунок 2.28 – Мгновенный поиск на странице

Следующая документация, на которую можно перейти – Речевой экран. На рисунке 2.29 изображена страница с речевым экраном.

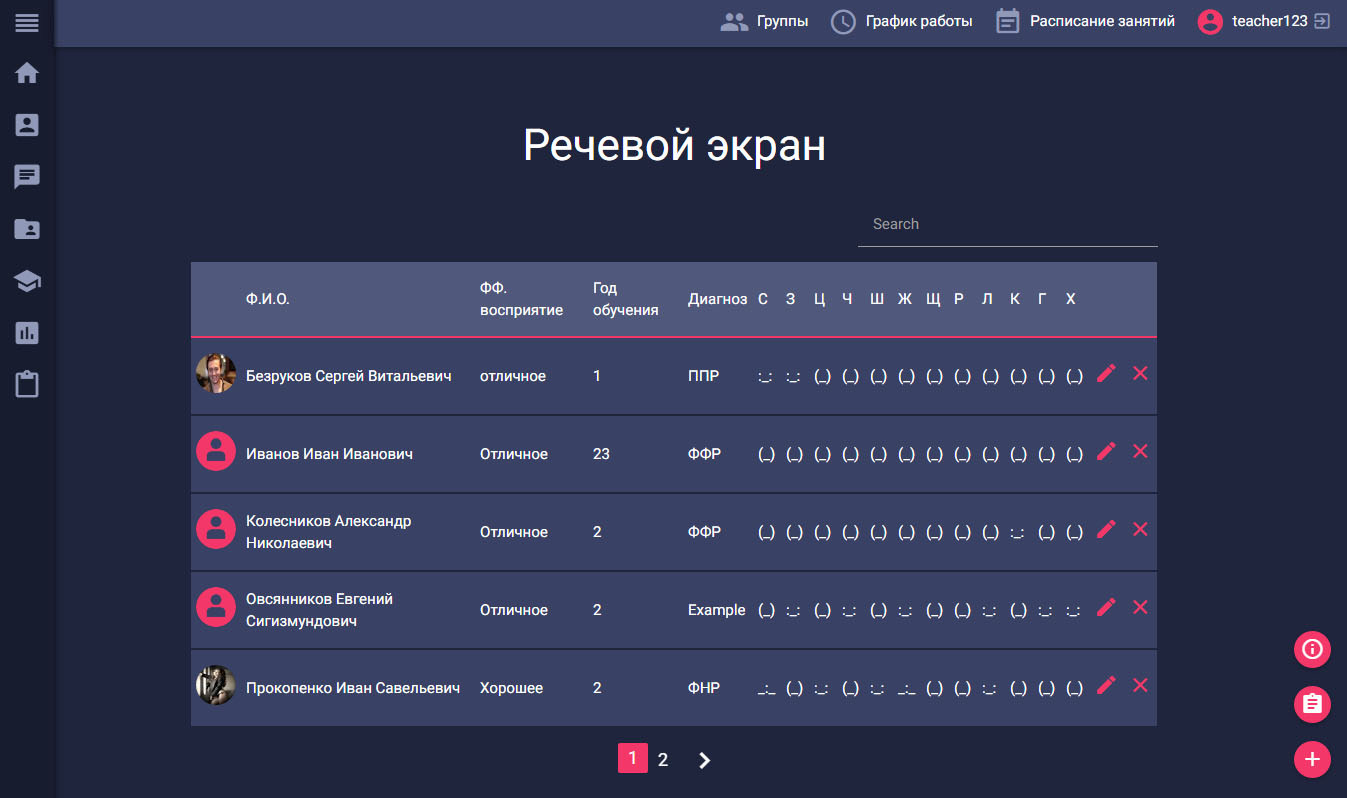


Рисунок 2.29 – Речевой экран

В этой документации также есть поиск по фамилии, кнопки редактирования, удаления и добавления, а также поиск и формирование отчета.

Обозначения, которые присутствуют для каждой буквы, обозначают текущее состояние звука у ребенка. Эту информацию можно посмотреть, нажав на кнопку в нижнем правом углу, рядом с уже знакомыми кнопками добавления и формирования отчета. Откроется модальное окно с обозначениями, которое показано на рисунке 2.30.

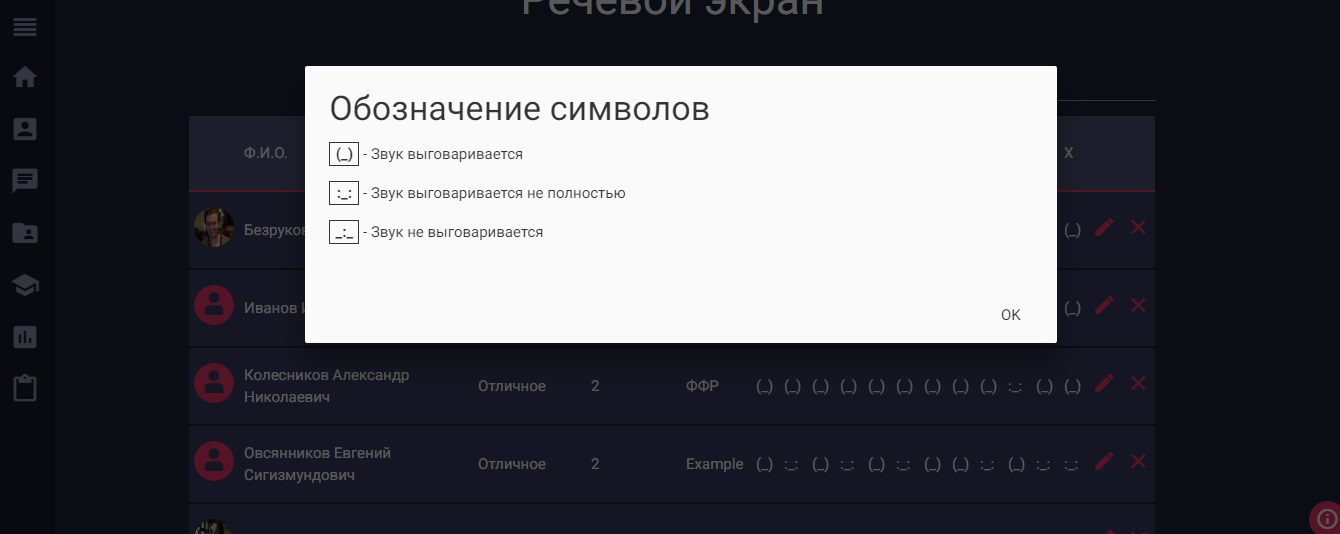


Рисунок 2.30 – Модальное окно с обозначением.

Перейдя по документации «Индивидуальный план», откроется окно со списком детей, для которых есть индивидуальный план коррекционной работы. На рисунке 2.31 изображен список детей с планом для каждого из них.

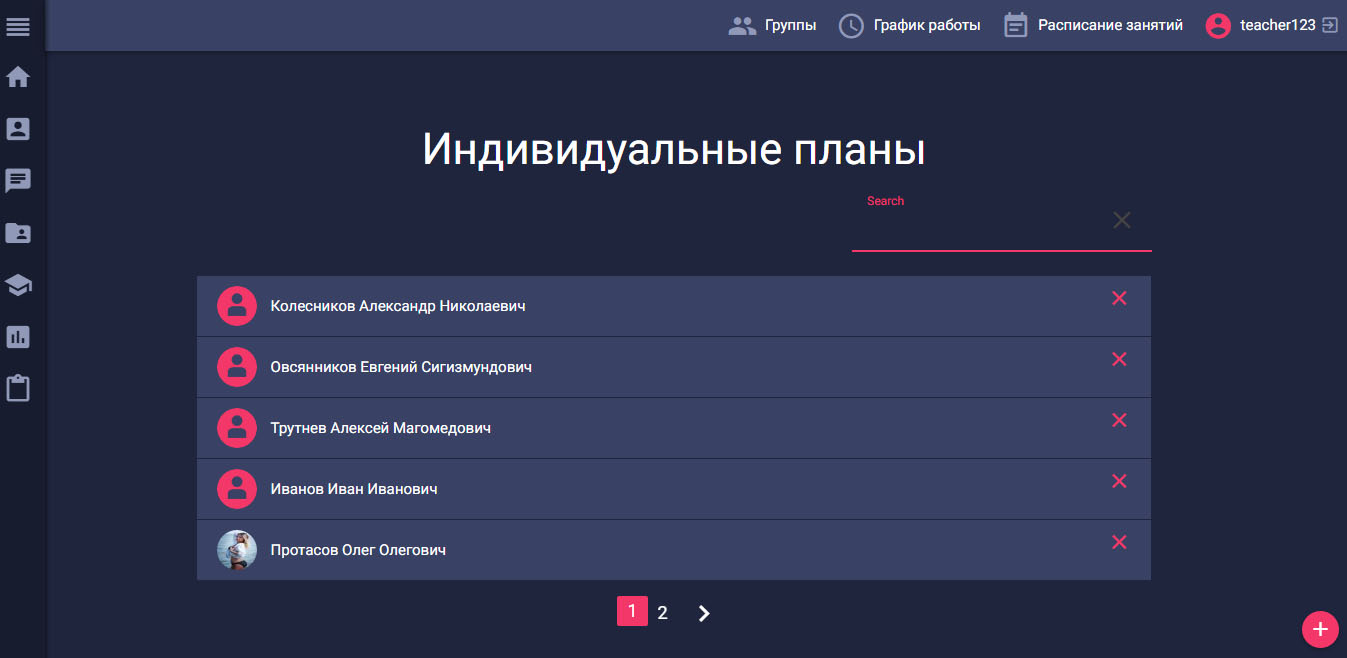


Рисунок 2.31 – Список детей для которых есть план.

На данной странице есть кнопки удаления, добавления, а также если нажать на ФИО ребенка, откроется страница редактирования плана ребенка. На рисунке 2.32 показана страница редактирования плана.

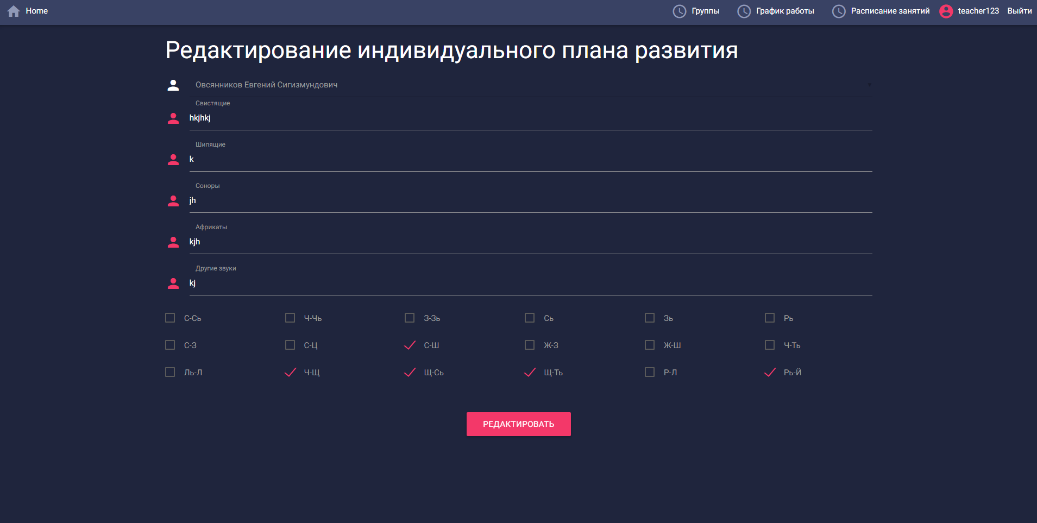


Рисунок 2.32 – Редактирование индивидуального плана.

Следующая документация – Карта речевого развития. На рисунке 2.33 изображена страница со списком детей для которых есть карта речевого развития.

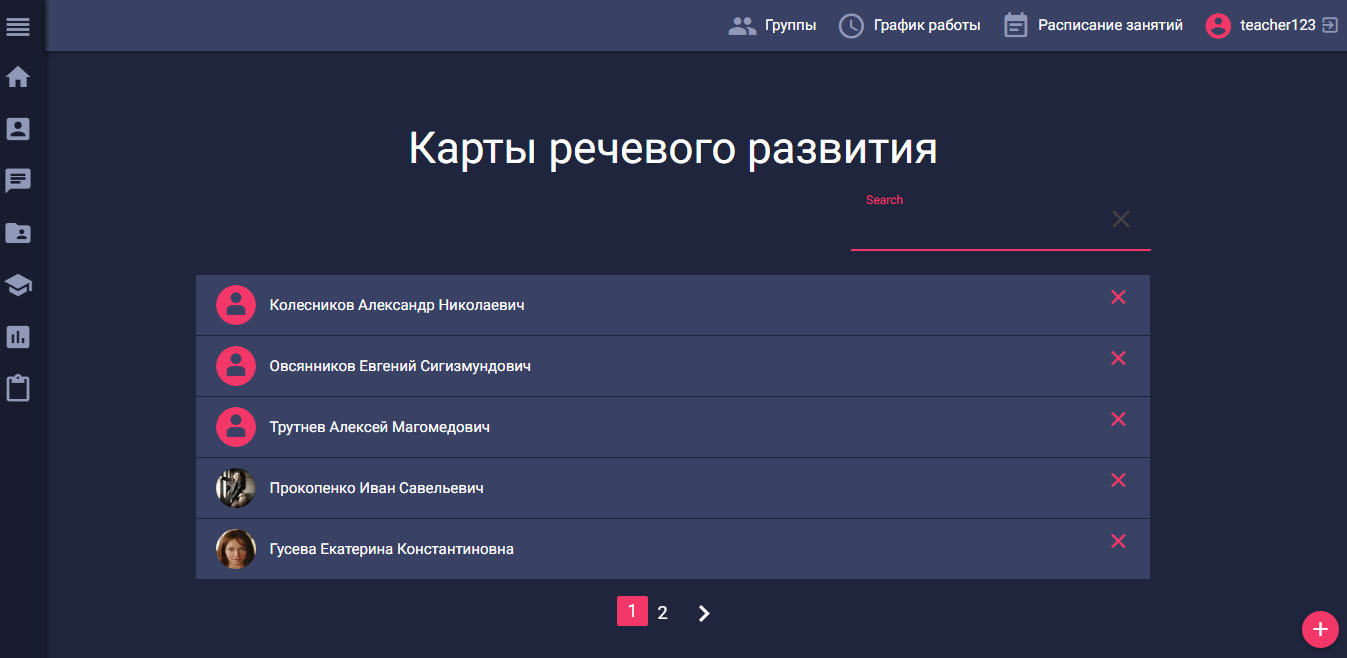


Рисунок 2.33 – Список детей у которых есть карта речевого развития.

На данной странице есть кнопки удаления, добавления, а также если нажать на ФИО ребенка, откроется страница редактирования карты ребенка. На рисунке 2.34 показана страница редактирования карты.

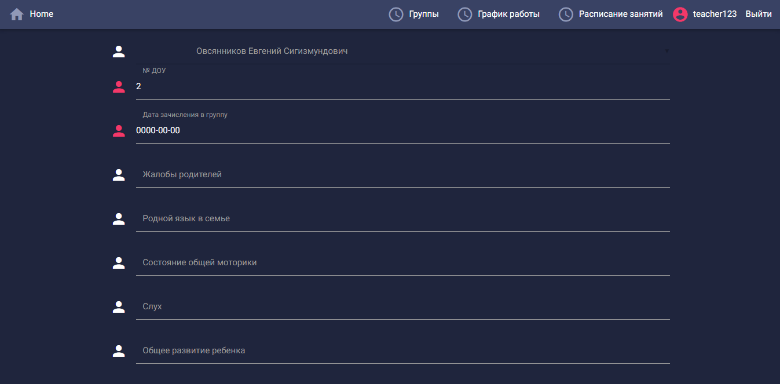


Рисунок 2.34 – Страница редактирования карты речевого развития.

Следующая документация, предназначена для ведения анализа результативности. На рисунке 2.35 изображена страница анализа результативности.

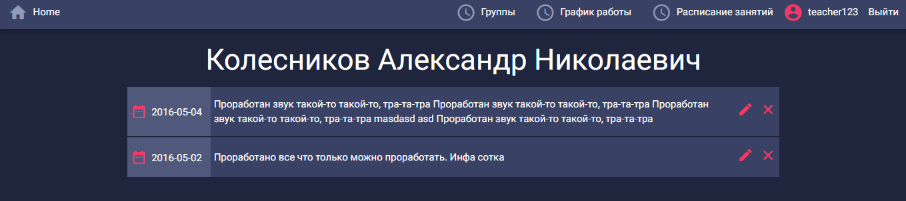


Рисунок 2.35 – Анализ результативности.

На странице присутствуют кнопки удаления, редактирования, и добавления новой заметки. А также дата, которая устанавливается автоматически при добавлении новой заметки.

Следующая и последняя документация – Индивидуальная карта развития. На рисунке 2.36 изображена страница с индивидуальной картой.

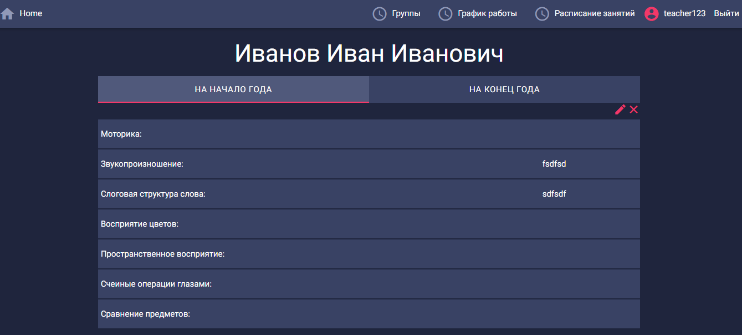


Рисунок 2.36 – Индивидуальная карта развития.

На данной странице показана карта развития на начало и конец года. Присутствуют кнопки удаления и редактирования.

Также, есть возможность создать, редактировать или удалять группы детей. На рисунке 2.37 изображена страница с группами.

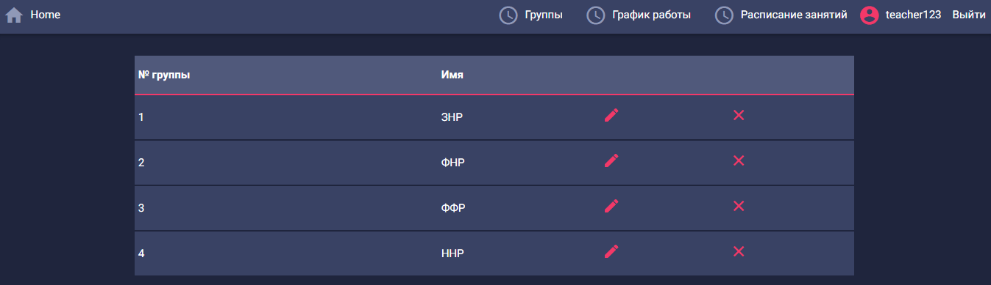


Рисунок 2.37 – Группы детей

Чтобы выйти из аккаунта, необходимо нажать на название своего аккаунта в верхнем меню справа. При этом откроется модальное окно о подтверждении, выхода которое показано на рисунке 2.38.

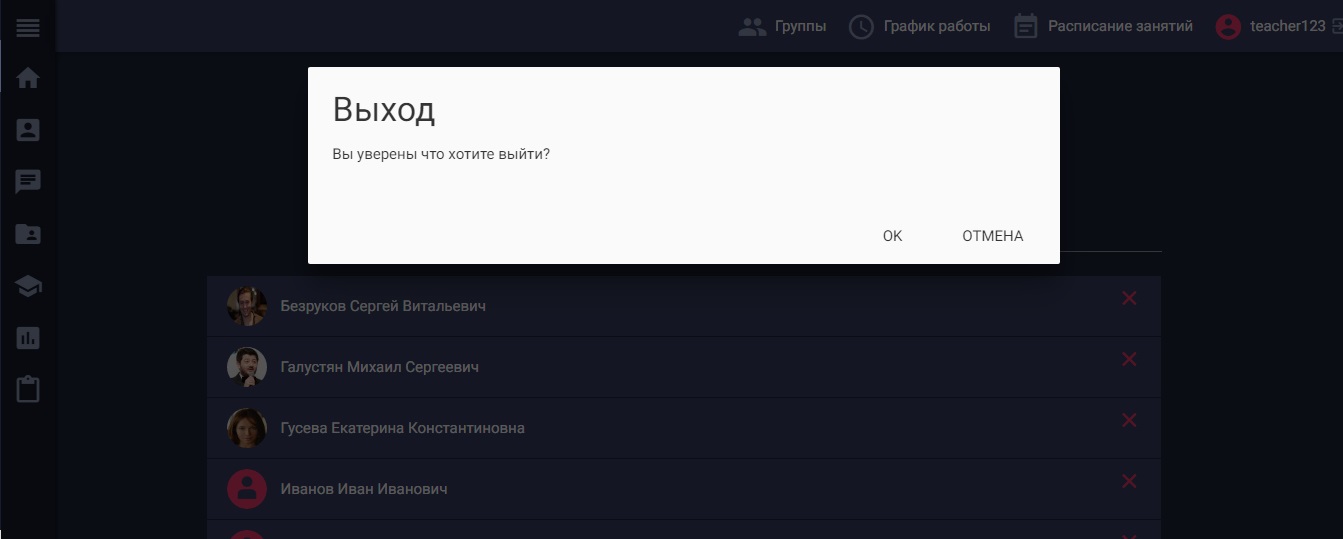


Рисунок 2.38 – Модальное окно подтверждения выхода

* + 1. Инструкция администратора

Если авторизоваться на сайте как администратор, то откроется главная страница администратора, где можно редактикровать аккаунты, менять группу пользователей, добавлять новые профили и многое другое. На рисунке 2.39 показана главная страница панели администратора.

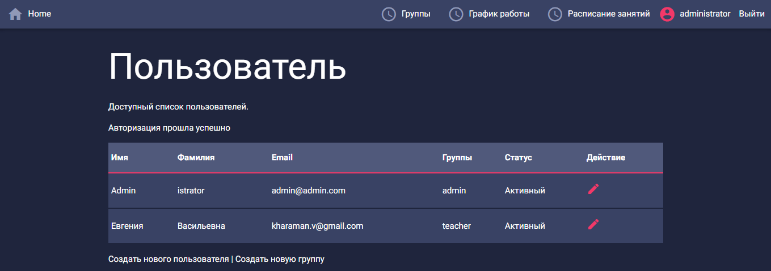


Рисунок 2.39 – Панель администратора

Если нажать на имя группы в соответствующей колонке, откроется страница редактирования группы, которое показано на рисунке 2.40.

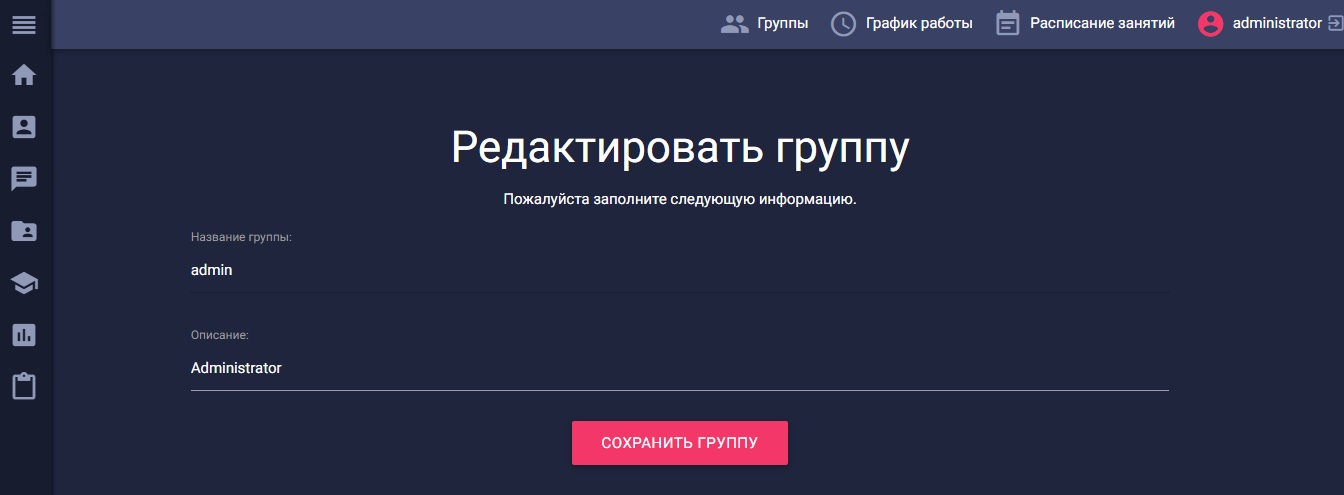


Рисунок 2.40 – Страница редактирования группы

Также, можно редактировать сам профиль, это показано на рисунке 2.41.

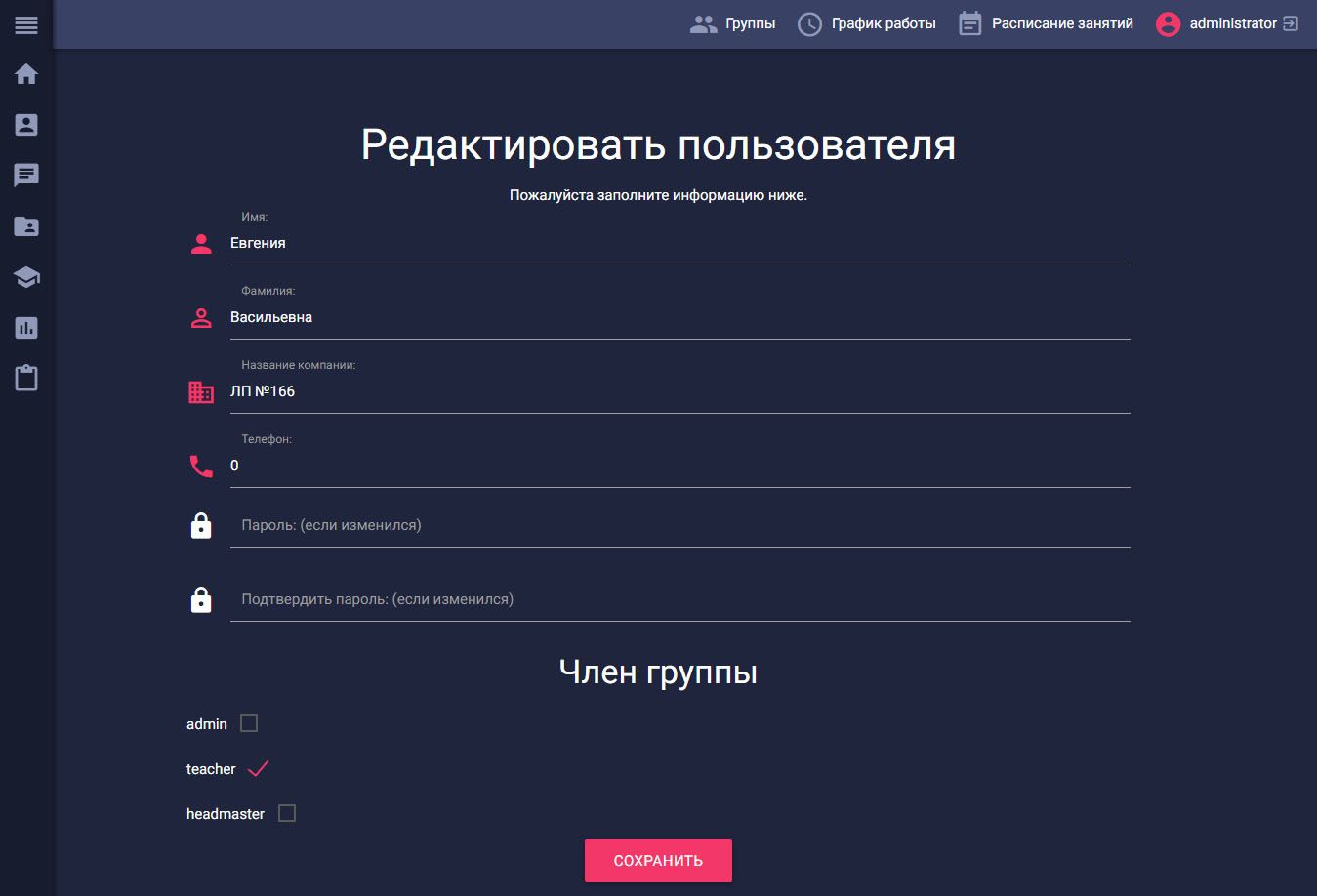


Рисунок 2.41 – Страница редактирования пользователя

Если необходимо добавить пользователя, то необходимо нажать на кнопку добавления в правом нижнем углу. Откроется страница добавления которая показана на рисунке 2.42.

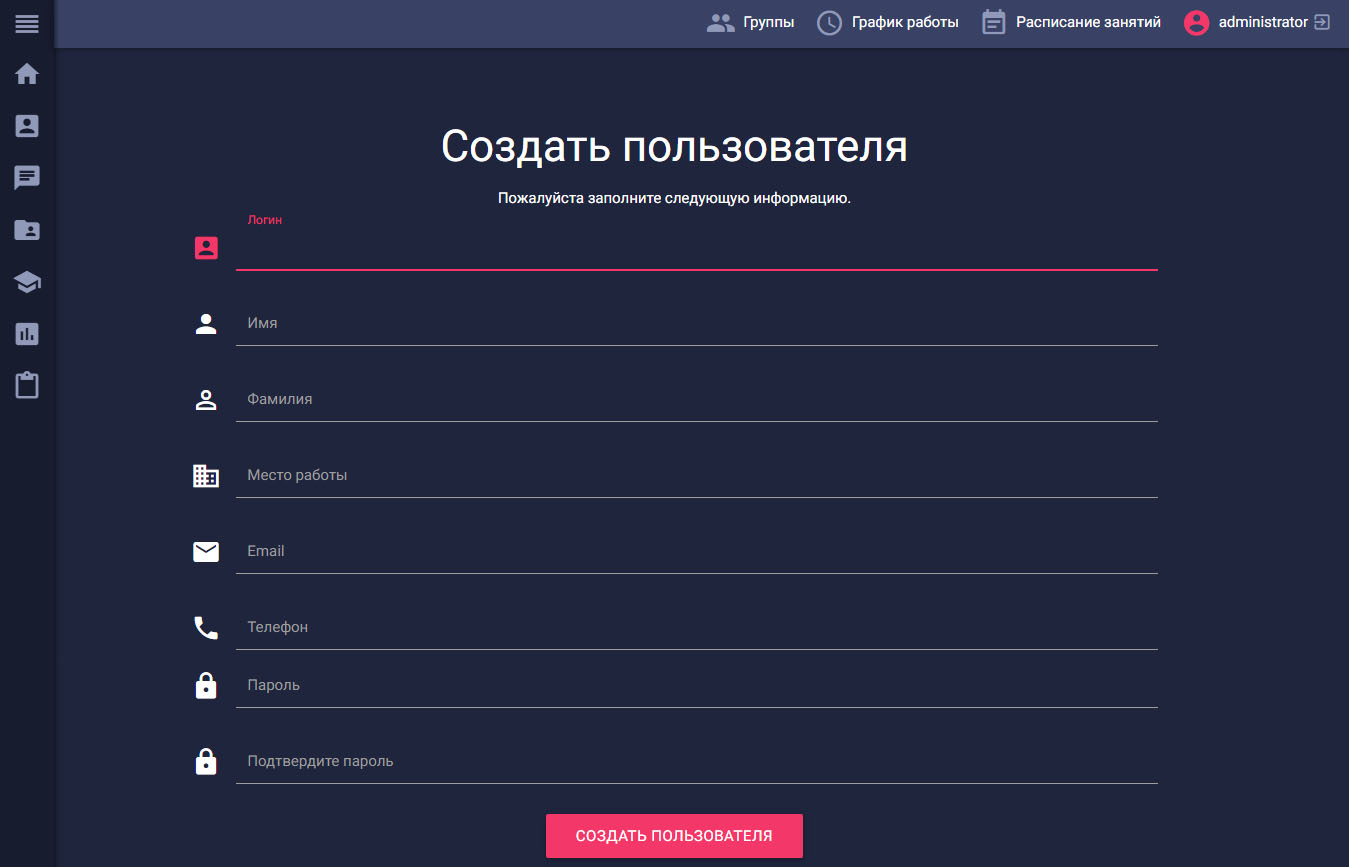


Рисунок 2.42 – Страница добавления нового пользователя

По такому же принципу, можно добавить группу. Это показано на рисунке 2.43.

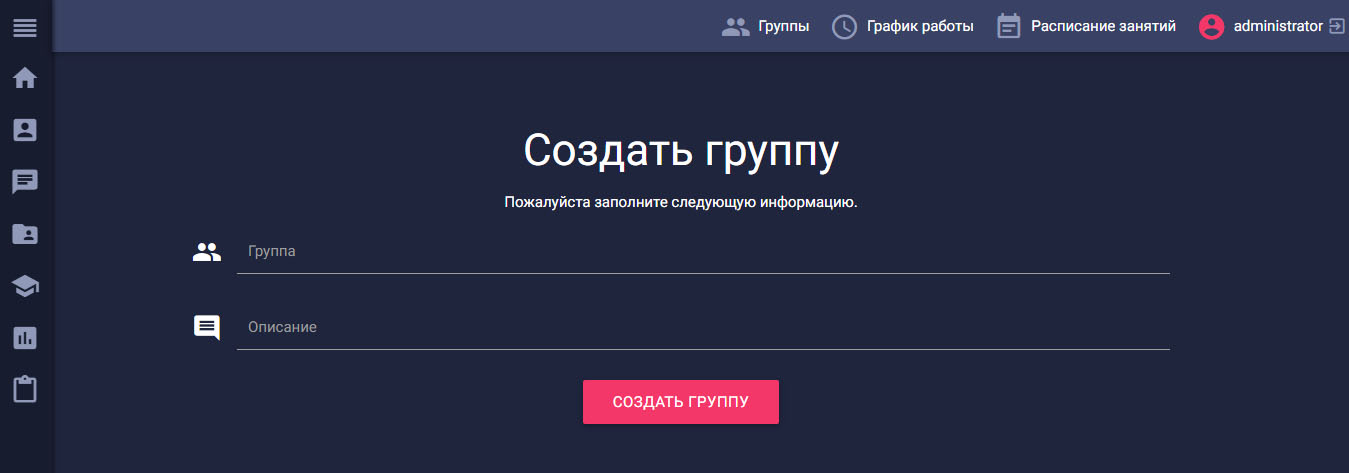


Рисунок 2.43 – Страница добавления новой группы

* + 1. Другие инструкции (специалиста, программиста, системного программиста)

Если зайти на сайт как «Заведующая ПМПК», то можно будет редактировать и добавлять график работы для учителя-логопеда. На рисунке 2.44 изображен график работы.

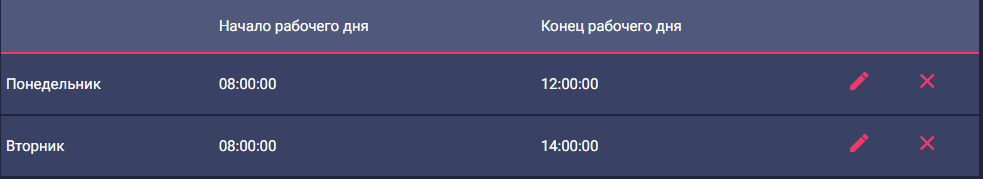


Рисунок 2.44 – График работы

Редактировать график может только заведующая ПМПК, а учитель-логопед может только просматривать график.

* 1. Техническое обеспечение
     1. Обоснование выбора (разработки) технического обеспечения АИС

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением пользователем совокупности организационно - технических мероприятий, перечисленных ниже:

– организацией бесперебойного питания технических средств;

– использованием лицензионного программного обеспечения;

– регулярным выполнением рекомендаций о типовые нормы времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождения программных средств;

– регулярным выполнением требований по защите информации и испытаниях программных средств на наличие компьютерных вирусов.

Время восстановления после отказа, вызванного збием электропитания технических средств (другими внешними факторами), не является фатальным збием (не крах) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств. Время восстановления после отказа, вызванного неисправностью технических средств, фатальным збием (крахом) операционной системы, не должно превышать времени, необходимого на устранение неисправностей технических средств и переустановки программных средств.

Отказа программы возможны в результате некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

С разрабатываемой системой должно иметь возможность работать десять пользователей, поэтому должна быть выбрана клиент-серверная технология хранения и обработки информации.

СУБД должна обеспечивать возможность работы в многопользовательском режиме, восстановления работоспособности при программно - аппаратных збиях, таких как отключение электропитания, проблемы с аппаратным обеспечением, нештатное завершение работы.

С целью обеспечения надежного функционирования в СУБД должны быть предусмотрены:

– сохранение работоспособности системы при некорректных операциях пользователя (ввод некорректных данных);

– сохранение целостности данных при нештатном завершении работы системы;

– резервное копирование данных;

– журнала операционной системы.

Данным требованиям отвечают такие СУБД, как, SQL Server, Oracle, InterBase, Firebird, MySQL. Выбор других СУБД нецелесообразен. [3]

Требования к серверу: процессор Intel Core, минимальный объем оперативной памяти 2 Гб, минимальный объем дискового пространства 60 ГБ, операционная система Windows XP / Vista / 7.

Требования к клиентской машине: процессор Intel Core, минимальный объем оперативной памяти 2 ГБ, минимальный объем дискового пространства 60 ГБ, операционная система Windows XP / Vista / 7. И периферийное устройство - принтер.

Разрабатывается система должна предусматривать возможность дальнейшего развития, модификации и включения новых функций в систему, улучшение кода, возможность расширения механизма аутентификации.

В качестве объектно-ориентированного языка программирования могут быть выбраны такие как, PHP, C ++, C#. Выбор других языков нецелесообразен.

Системные программные средства, используемые программой, должны быть представлены лицензионной версии операционной системы Windows ХР и выше, MS Office, Firebird 2.5 или выше.

В процессе работы программы входной информацией для программы должны быть: файлы баз данных, манипуляции мышью, а также вводимая пользователем на клавиатуре ЭВМ, согласно режимам, определяемых выходной экранной информацией.

Так как информационная система будет многопользовательской, то в ней будут использоваться разграничения доступа к системе, а вход в систему будет происходить по паролю.

* + 1. Структура сетевой системы (топологическая и логическая)

В основе проектирования АС лежит клиент-серверная архитектура.

Клиент-сервер представляет собой вычислительную или сетевую архитектуру, в которой задания или сетевая нагрузка распределены между поставщиками услуг, которые называются серверами, и заказчиками услуг, называемых клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением.

На рисунке 2.45 изображена структура системы.



Рисунок 2.45 - Структура сетевой системы (топологическая и логіческая)

На диаграмме отображены компоненты и их взаимосвязь.

* + 1. Структура комплекса средств автоматизации

Комплекс способов автоматизации (КСА), исполняет ряд функций по обработке информации. Система включает в себя автоматизацию учета работы учителя-логопеда.

Такую, как разработка системы, которая поможет учителю-логопеду в обработке, учете и автоматизации деятельности. Увеличение эфективности в работе, в плане комфорта и скорости.

* + 1. Спецификация оборудования

Требования к серверу: процессор Intel Core, минимальный объем оперативной памяти 2 Гб, минимальный объем дискового пространства 60 ГБ, операционная система Windows XP / Vista / 7. Требования к клиентской машине: процессор Intel Core, минимальный объем оперативной памяти 2 ГБ, минимальный объем дискового пространства 60 ГБ, операционная система Windows XP / Vista / 7. И периферийное устройство - принтер.

3 ОХРАНА ТРУДА

3.1 Опасные, вредные факторы и пожароопасность усовершенствованного технологического процесса при использовании результатов исследования в производстве.

3.1.1 Общие сведения о предприятии

Психолого-медико-педагогические консультации являются методическими учреждениями системы образования и науки Украины, осуществляющими консультативную, методическую, психолого-педагогическую и профилактическую деятельность.

В консультации проводится диагностика интеллектуальных способностей ребенка, его психической зрелости и готовности к школьному обучению; выявление индивидуальных особенностей развития, личностных особенностей и возможностей ребенка к обучению; разработка рекомендаций по уточнению, а в необходимых случаях и изменению программы обучения; проведение медицинской диагностики состояния соматического и нервно-психического здоровья; проведение компьютерных энцефалографии и эхо-энцефалографии.

Учитель-логопед находится на логопедическом пункте (дет. сад и т.д.), где и нужно учитывать вредные и опасные факторы.

3.1.2 Опасные и вредные факторы

Поскольку данная работа подразумевает длительное пребывание за компьютером, отсюда вытекают различные вредные факторы, такие как:

* повышенная температура поверхностей ПК;
* повышенная или пониженная температура воздуха рабочей зоны;
* выделение в воздух рабочей зоны ряда химических веществ;
* повышенная или пониженная влажность воздуха;
* повышенный или пониженный уровень отрицательных и положительных аэроионов;
* повышенное значение напряжения в электрической цепи, замыкание;
* повышенный уровень статического электричества;
* повышенный уровень электромагнитных излучений;
* повышенная напряженность электрического поля;
* отсутствие или недостаток естественного света;
* недостаточная искусственная освещенность рабочей зоны;
* прямая и отраженная блесткость;
* зрительное напряжение;
* монотонность трудового процесса;
* нервно-эмоциональные перегрузки.

**Работа на ПК сопровождается постоянным и значительным напряжением функций зрительного анализатора**. Одной из основных особенностей является иной принцип чтения информации, чем при обычном чтении.

При обычном чтении текст на бумаге, расположенный горизонтально на столе, считывается работником с наклоненной головой при падении светового потока на текст.

При работе на ПК пользователь считывает текст, почти не наклоняя голову, глаза смотрят прямо или почти прямо вперед, текст (источник — люминесцирующее вещество экрана) формируется по другую сторону экрана, поэтому пользователь не считывает отраженный текст, а смотрит непосредственно на источник света, что вынуждает глаза и орган зрения в целом работать в несвойственном ему стрессовом режиме длительное время.

Типичными ощущениями, которые испытывают к концу рабочего дня пользователи системы, являются: переутомление глаз, головная боль, тянущие боли в мышцах шеи, рук и спины, снижение концентрации внимания.

Уже в первые годы компьютеризации было отмечено специфическое зрительное утомление у пользователей дисплеев, получившее общее название «компьютерный зрительный синдром». Одной из причин служит то, что сформировавшаяся за миллионы лет эволюции зрительная система человека приспособлена для восприятия объектов в отраженном свете (печатные тексты, рисунки и т.п.), а не для работы за дисплеем.

Изображение на дисплее принципиально отличается от привычных глазу объектов наблюдения — оно светится, мерцает, состоит из дискретных точек, а цветное компьютерное изображение не соответствует естественным цветам.

Но не только особенности изображения на экране вызывают зрительное утомление. Большую нагрузку орган зрения испытывает при вводе информации, так как пользователь вынужден часто переводить взгляд с экрана на текст и клавиатуру, находящиеся на разном расстоянии и по-разному освещенные.

Зрительное утомление проявляется жалобами на затуманивание зрения, трудности при переносе взгляда с ближних предметов на дальние и с дальних на ближние, кажущиеся изменения окраски предметов, их двоение, чувство жжения, «песка» в глазах, покраснение век, боли при движении глаз.

Длительная и интенсивная работа на компьютере может стать источником тяжелых профессиональных заболеваний, таких, как травма повторяющихся нагрузок (ТПН), представляющая собой постепенно накапливающиеся недомогания, переходящие в заболевания нервов, мышц и сухожилий руки.

К профессиональным заболеваниям, связанным с ТПН, относятся:

* тендовагинит — воспаление сухожилий кисти, запястья, плеча;
* тендосиновит — воспаление синовиальной оболочки сухожильного основания кисти и запястья;
* синдром запястного канала (СЗК) – вызывается ущемлением срединного нерва в запястном канале. Накапливающаяся травма вызывает образование продуктов распада в области запястного канала, в результате чего вначале возникает отек, а затем СЗК.

Появляются жалобы на жгучую боль и покалывание в запястье, ладони, а также пальцах, кроме мизинца. Наблюдается болезненность и онемение, ослабление мышц, обеспечивающих движение большого пальца.

Эти заболевания обычно наступают в результате непрерывной работы на неправильно организованном рабочем месте.

3.2 Мероприятия по обеспечению безопасности усовершенствованной технологии (или оборудования) при использовании результатов исследования в производстве.

* + 1. Освещенность рабочего места.

К системам производственного освещения предъявляются следующие основные требования: соответствие уровня освещенности рабочих мест характеру выполняемой зрительной работы; достаточно равномерное распределение яркости на рабочих поверхностях и в окружающем пространстве; отсутствие резких теней, прямой и отраженной блескости; постоянство освещенности во времени; оптимальная направленность излучаемого осветительными приборами светового потока; долговечность, экономичность, электробезопасность и пожаробезопасность, удобство и простота эксплуатации. Нормы освещенности помещений на уровне 80 см от пола отражены в таблице 3.1.

Таблица 3.1 – Нормы освещенности помещений на уровне 80 см от пола.

|  |  |  |
| --- | --- | --- |
| Тип помещения | Нормы освещенности, лк при освещении | |
| Комбинированное освещение | Общее освещение |
| Компьютерный зал | 750 | 400 |
| Помещения для персонала, осуществляющего техническое обслуживание. | 750 | 400 |
| Архивы, помещения для хранения носителей информации | — | 300 |

В соответствии с нормами по освещению ниже перечислены требования.

– Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк.

– Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк.

– Следует ограничивать прямую блесткость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/кв.м.

– Следует ограничивать отраженную блесткость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ВДТ и ПЭВМ не должна превышать 40 кд/кв.м и яркость потолка, при применении системы отраженного освещения, не должна превышать 200 кд/кв.м.

– В качестве источников света при искусственном освещении должны применяться преимущественно люминесцентные лампы типа ЛБ. При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп мощностью до 250 Вт. Допускается применение ламп накаливания в светильниках местного освещения.

– Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

– Коэффициент запаса (Кз) для осветительных установок общего освещения должен приниматься равным 1.4.

* + 1. Допустимые значения параметров излучений, генерируемых мониторами.

В таблице 3.2 указаны допустимые значения излучений, генерируемых мониторами

Таблица 3.2 – допустимые значения излучений, генерируемых мониторами.

|  |  |
| --- | --- |
| Мощность экспозиционной дозы рентгеновского излучения на расстоянии 0,05 м вокруг видеомонитора | 100 мкР/час |
| Электромагнитное излучение на расстоянии 0,5 м вокруг монитора  по электрической составляющей: | |
| 5 Гц - 2 кГц | 25 В/м |
| 2 – 400 кГц | 2,5 В/м |
| по магнитной составляющей: | |
| 5 Гц - 2 кГц | 250 нТл |
| 2 - 400 кГц | 25 нТл |
| Поверхностный электростатический потенциал | Не более 500 В |

На таблице отображены значения излучений, которое генерируются на мониторах.

3.2.3 Организация рабочего места в соответствии с требованиями эргономики.

Под рабочим местом условно понимают зону, оснащенную необходимыми техническими средствами, где работник или группа работников постоянно или временно выполняют одну работу или операцию.

Правильная планировка рабочих мест может существенно снизить действие целого ряда неблагоприятных факторов, действующих на работающего с ПЭВМ человека.

Санитарными правилами предусмотрены следующие нормы.

* + помещения с ВДТ (видео-дисплейные терминалы) и ЭВМ должны иметь естественное и искусственное освещение.
  + естественное освещение должно осуществляться через светопроемы, ориентированные преимущественно на север и северо-восток и обеспечивать коэффициент естественной освещенности (КЕО) не ниже 1.2.
  + для внутренней отделки интерьера помещений с ВДТ и ЭВМ должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка - 0.7 - 0.8; для стен - 0.5 - 0.6; для пола - 0.3 - 0.5.
  + конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ВДТ и ЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.
  + рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.
  + экран видеомонитора должен находиться от глаз пользователя на оптимальном расстоянии 600 - 700 мм, но не ближе 500.
  + рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.
  + клавиатуру следует располагать на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

Схемы размещения рабочих мест с ВДТ и ЭВМ должны учитывать расстояния между рабочими столами с видеомониторами (в направлении тыла поверхности одного видеомонитора и экрана другого видеомонитора), которое должно быть не менее 2,0 м, а расстояние между боковыми поверхностями видеомониторов - не менее 1,2 м.

Оконные проемы в помещениях использования ВДТ и ЭВМ должны быть оборудованы регулируемыми устройствами типа: жалюзи, занавесей, внешних козырьков и др.

Кроме этого существуют ряд отдельных требований к организации и оборудованию рабочих мест для взрослых пользователей, для учащихся средних и высших учебных заведений, а также детей дошкольного возраста, которые учитывают специфику каждой из этих групп.

3.2.4 Рациональная организация режима труда и отдыха.

Для предупреждения заболеваний, связанных с работой на компьютере необходима рациональная организация труда и отдыха, которая нормируется в соответствии с санитарными правилами.

Режимы труда и отдыха при работе с ЭВМ и ВДТ должны организовываться в зависимости от вида и категории трудовой деятельности.

Для обеспечения оптимальной работоспособности, сохранения здоровья пользователей, а также предупреждения развития переутомления на протяжении рабочей смены должны устанавливаться регламентированные перерывы, которые указаны в таблице 3.3.

Таблица 3.3 – Регламентирующие перерывы.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Категория работы с ВДТ или ЭВМ | Уровень нагрузки за рабочую смену при видах paбoт с ВДТ | | | Суммарное время регламентированных перерывов, мин. | |
| Группа А, количество знаков | группа Б, количество знаков | Группа В, часов | при 8-ми часовой рабочей смене | При 12-ти часовой рабочей смене |
| I | До 26000 | до 15000 | до 2,0 | 30 | 70 |
| II | До 40000 | до 30000 | до 4,0 | 50 | 90 |
| III | До 60000 | до 40000 | до 6,0 | 70 | 120 |

На таблице отображены перерывы в работе.

Выводы

В ходе выполнения дипломного проекта была достигнута основная цель работы – разработана автоматизированная информационная система учёта работы учителя-логопеда. А именно автоматизирован учет работы документации.

Для достижения поставленной цели были проведены изучение и анализ уже существующей структуры автоматизированных систем. При этом был выявлен ряд недостатков существующей системы, которые возможно устранить за счет разработки и внедрения новой АИС.

Для осуществления разработки были использованы следующие технологии:

* для создания базы данных, хранящей необходимую для функционирования АИС, информацию о детях в логопедических пунктах, и последующей работы с ней была выбрана СУБД MySQL. MySQL отличатся хорошей скоростью работы, надежностью, гибкостью и при этом распространяется совершенно бесплатно;
* программное обеспечение для работы с MySQL было выбрано HeidiSQL, предоставляющее возможность быстрого управления и конфигурирования баз данных;
* в качестве языка программирования был выбран php, предоставляющий возможность создания различного рода веб-приложений, управляемых базами данных, и поддерживающий СУБД MySQL;
* в качестве фреймворка был выбран фреймворк Codeigniter 3, идеально подходящий для целей дипломного проекта.

ПЕРЕЧЕНЬ ССЫЛОК

1. Конверс Т. РНР 5 и MySQL. Библия пользователя / Т. Конверс. – М. : Вильямс, 2006. – 1216 с.

2. Квинт И. HTML, XHTML и CSS на 100%. /И. Квинт. – СПб. : Питер, 2010. – 384 с. : ил.

3. Алексеев А.П. Введение в Web-дизайн: учебное пособие. /А.П. Алексеев. – М. : СОЛОН-ПРЕСС, 2008. – 61 с. : ил.

4. [Кацуба Ольга](http://www.e-reading.club/bookbyauthor.php?author=42546) Психолого-медико-педагогические консультации / [Кацуба Ольга](http://www.e-reading.club/bookbyauthor.php?author=42546) :  «Ай Пи Эр Медиа» – 284 с. : ил.

5. Зелинский С. Excel 2003 / С. Зелинский. – СПб. : Лидер, 2005. – 496 с. : ил.

6. ОРММ ИСЖТ 2.01-00. Требования к составу, содержанию и оформлению документов при создании ИС. – М. : ВНИИАС МПС России, 2000. – 62 с.

7. Баричев С. Г. Основы работы педагогических консультаций / С. Г. Баричев, Р. Е. Серов. – М. : МИФИ, 2007. – 482 с.

8. Минитаева А. М. Разработка и стандартизация программных средств и информационных технологий: учеб. пособие / А. М. Минитаева. – Омск : ОмГТУ, 2011. – 92 с.

9. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для ВУЗов / В. Г. Олифер, Н. А.Олифер. – Изд. 3-е. – СПб. : Питер, 2006. – 958 с. : ил.

10. Городские службы обчуения [Электронный ресурс]. – Режим доступа: http://www.cityservices.com.ua/tour/

11. Логопедия [Электронный ресурс]. – Режим доступа: https://www.logos.ru

12. Гвоздева В. А. Основы построения автоматизированных информационных систем : учебник / В. А. Гвоздева, Ю. И. Лаврентьева. – М. : ИД "Форум": ИНФРА-М, 2007. – 320 с.

13. Дубаков А. А. Проектирование информационных систем / А. А. Дубаков. – Томск : Изд. Томского политехнического университета, 2011. – 258 с.

14. Королев М. А. Информационные системы и структуры данных / М. А. Королев. – М. : Статистика, 2007. – 184 с.

15. Леффингуэлл Д. Принципы работы с требованиями к программному обеспечению. Унифицированный подход / Д. Леффингуэлл, Д. Уидриг ; пер. с англ. – М. : Вильяме, 2002. – 448 с.

16. Мацяшек Л. А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML / Л. А. Мацяшек ; пер. с англ. – М. : Издательский дом "Вильямс", 2002. – 432 с.

17. Грекул В. И. Проектирование информационных систем : учебн. пособ. / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – М. : БИНОМ. Лаборатория знаний, 2008. – 300 с.

18. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание. : Пер. с англ. – М.: Издательский дом "Вильямс", 2003. – 1440с. : ил.

19. Мухин В. И. Исследование систем управления : учебник для вузов / В. И. Мухин – М. : Изд. "Экзамен", 2003. – 384 с.

20. Калянов, Г. Н. CASE. Структурный системный анализ (автоматизация и применение) / Г. Н. Калянов. – М. : Изд-во "Лори", 1996.

21. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose / А. В. Леоненков. – М. : ИНТУИТ.РУ, 2013. – 320 с.

22. Кравец О. Я. Практикум по проектированию информационных систем : учебное пособие / О. Я. Кравец, С. А. Олейникова. – Изд. 2-е, перераб. и доп. – Воронеж : Научная книга, 2006. – 208 с.

23. Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс. – М.: Вильямс, 2003. – 1088 с.

24. Томсон Л. Разработка Web-приложений на PHP и MySQL / Л. Томсон, Л. Веллинг. – К.: ДиаСофт, 2001. – 672 с.]

25. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования /Э. Гамма, Р. Хелм, Р. Джонсон, Влисидес Дж. – СПб: Питер, 2010. – 368 с.

26. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебник для ВУЗов / В. Г. Олифер, Н. А. Олифер. – 3-е изд. – СПб. : Питер, 2006. – 958 с. : ил.

27. Барановская Т. П. Архитектура компьютерных систем и сетей : учеб. пособие / Т. П. Барановская, В. И. Лойко, М. И. Семенов, А. И. Трубилин; под ред. В. И. Лойко. – М. : Финансы и статистика, 2003. – 256 с. : ил.

28. Карпов Д. В. Гибкая методология разработки программного обеспечения / Вестник Нижегородского университета им. Н.И. Лобачевского, 2011, No 3 (2), С. 227–230.

29. Анкудинов Г. И. Сети ЭВМ и телекоммуникации. Архитектура и сетевые технологи : учеб. пособие / Г. И. Анкудинов, И. Г. Анкудинов, А. И. Стрижаченко. – СПб. : СЗТУ, 2006. – 182 с.

30. Безопасность жизнедеятельности [Электронный ресурс] : курс лекций по дисциплине «Безопасность жизнедеятельности» для студентов всех специальностей и форм обучения / сост. : О. Н. Аксёнова, В. А. Бурко, Н. Ю. Елистратова, О. Ю. Нестеров, В. Л. Черепня, М. В. Ярмонова. – Мариуполь : ПГТУ, 2012. – 182 с. – Режим доступа : http://umm.pstu.edu/handle/123456789/510

Приложение А

Листинг

Analysis.php

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Analysis extends CI\_Controller

{

public $limit = 5;

public function \_\_construct()

{

parent::\_\_construct();

if (!$this->ion\_auth->logged\_in()) {

redirect('auth/login');

exit;

}

$this->load->model('analysis\_model', 'analysis');

}

public function search()

{

$data['children'] = $this->analysis->search($this->input->get('q'));

$data['pagination'] = '';

$view['title'] = 'Анализ результативности';

$this->load->view('header', $view);

$this->load->view('analysis/index', $data);

$this->load->view('footer');

}

public function ajax\_search()

{

$data['children'] = $this->analysis->search($this->input->get('q'));

$data['controller'] = 'analysis';

$this->load->view('ajax\_search', $data);

}

public function index()

{

$config = [

'base\_url' => '/analysis/index/',

'total\_rows' => $this->analysis->count(),

'per\_page' => $this->limit,

'use\_page\_numbers' => TRUE

];

$this->pagination->initialize($config);

$data['children'] = $this->analysis->get\_added\_children($this->limit);

$data['pagination'] = $this->pagination->create\_links();

$view['title'] = 'Анализ результативности';

$this->load->view('header', $view);

$this->load->view('analysis/index', $data);

$this->load->view('footer');

}

public function view($children\_id)

{

$data['analysis'] = $this->analysis->get\_children\_analysis($children\_id);

$data['child'] = $this->children->get\_child\_full\_name($children\_id);

$view['title'] = 'Анализ результативности';

$this->load->view('header', $view);

$this->load->view('analysis/view', $data);

$this->load->view('footer');

}

public function create()

{

if ($this->form\_validation->run('analysis') == FALSE)

{

$data['children'] = $this->children->get\_children\_full\_names();

$view['title'] = 'Создание - Анализ результативности';

$this->load->view('header', $view);

$this->load->view('analysis/add', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('children\_id, description');

$data['date'] = date('Y-m-h');

$children\_id = $data['children\_id'];

if ($this->analysis->create($data))

{

redirect('/analysis/view/' . $children\_id);

exit;

}

}

}

public function edit($id)

{

if ($this->form\_validation->run('analysis') == FALSE)

{

$data['analysis'] = $this->analysis->get\_one($id);

$view['title'] = 'Редактирование - Анализ результативности';

$this->load->view('header', $view);

$this->load->view('analysis/edit', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('description');

$children\_id = $this->input->post('children\_id');

if ($this->analysis->edit($id, $data))

{

header('Location: /analysis/view/' . $children\_id);

exit;

}

}

}

public function delete($children\_id, $id)

{

if ($this->analysis->delete($id))

{

redirect('/analysis/view/' . $children\_id);

exit;

}

}

public function delete\_all($children\_id)

{

if ($this->analysis->delete\_all($children\_id))

{

redirect('/analysis');

exit;

}

}

}

Auth.php

<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Auth extends CI\_Controller {

function \_\_construct()

{

parent::\_\_construct();

$this->load->helper(array('url','language'));

$this->form\_validation->set\_error\_delimiters($this->config->item('error\_start\_delimiter', 'ion\_auth'), $this->config->item('error\_end\_delimiter', 'ion\_auth'));

$this->lang->load('auth');

}

// redirect if needed, otherwise display the user list

function index()

{

if (!$this->ion\_auth->logged\_in())

{

// redirect them to the login page

redirect('auth/login', 'refresh');

}

elseif (!$this->ion\_auth->is\_admin()) // remove this elseif if you want to enable this for non-admins

{

// redirect them to the home page because they must be an administrator to view this

//return show\_error('You must be an administrator to view this page.');

redirect('/home');

}

else

{

// set the flash data error message if there is one

$this->data['message'] = (validation\_errors()) ? validation\_errors() : $this->session->flashdata('message');

//list the users

$this->data['users'] = $this->ion\_auth->users()->result();

foreach ($this->data['users'] as $k => $user)

{

$this->data['users'][$k]->groups = $this->ion\_auth->get\_users\_groups($user->id)->result();

}

$view['title'] = 'Панель администратора';

$this->load->view('header', $view);

$this->\_render\_page('auth/index', $this->data);

$this->load->view('footer');

}

}

// log the user in

function login()

{

$this->data['title'] = $this->lang->line('login\_heading');

//validate form input

$this->form\_validation->set\_rules('identity', str\_replace(':', '', $this->lang->line('login\_identity\_label')), 'required');

$this->form\_validation->set\_rules('password', str\_replace(':', '', $this->lang->line('login\_password\_label')), 'required');

if ($this->form\_validation->run() == true)

{

// check to see if the user is logging in

// check for "remember me"

$remember = (bool) $this->input->post('remember');

if ($this->ion\_auth->login($this->input->post('identity'), $this->input->post('password'), $remember))

{

//if the login is successful

//redirect them back to the home page

$this->session->set\_flashdata('message', $this->ion\_auth->messages());

redirect('/', 'refresh');

}

else

{

// if the login was un-successful

// redirect them back to the login page

$this->session->set\_flashdata('message', $this->ion\_auth->errors());

redirect('auth/login', 'refresh'); // use redirects instead of loading views for compatibility with Y\_Controller libraries

}

}

else

{

// the user is not logging in so display the login page

// set the flash data error message if there is one

$this->data['message'] = (validation\_errors()) ? validation\_errors() : $this->session->flashdata('message');

$this->data['identity'] = array('name' => 'identity',

'id' => 'identity',

'type' => 'text',

'value' => $this->form\_validation->set\_value('identity'),

);

$this->data['password'] = array('name' => 'password',

'id' => 'password',

'type' => 'password',

);

$view['title'] = 'Авторизация';

$this->\_render\_page('auth/login', $this->data);

}

}

// log the user out

function logout()

{

$this->data['title'] = "Logout";

// log the user out

$logout = $this->ion\_auth->logout();

// redirect them to the login page

$this->session->set\_flashdata('message', $this->ion\_auth->messages());

redirect('auth/login', 'refresh');

}

// change password

function change\_password()

{

$this->form\_validation->set\_rules('old', $this->lang->line('change\_password\_validation\_old\_password\_label'), 'required');

$this->form\_validation->set\_rules('new', $this->lang->line('change\_password\_validation\_new\_password\_label'), 'required|min\_length[' . $this->config->item('min\_password\_length', 'ion\_auth') . ']|max\_length[' . $this->config->item('max\_password\_length', 'ion\_auth') . ']|matches[new\_confirm]');

$this->form\_validation->set\_rules('new\_confirm', $this->lang->line('change\_password\_validation\_new\_password\_confirm\_label'), 'required');

if (!$this->ion\_auth->logged\_in())

{

redirect('auth/login', 'refresh');

}

$user = $this->ion\_auth->user()->row();

if ($this->form\_validation->run() == false)

{

// display the form

// set the flash data error message if there is one

$this->data['message'] = (validation\_errors()) ? validation\_errors() : $this->session->flashdata('message');

$this->data['min\_password\_length'] = $this->config->item('min\_password\_length', 'ion\_auth');

$this->data['old\_password'] = array(

'name' => 'old',

'id' => 'old',

'type' => 'password',

);

$this->data['new\_password'] = array(

'name' => 'new',

'id' => 'new',

'type' => 'password',

'pattern' => '^.{'.$this->data['min\_password\_length'].'}.\*$',

);

$this->data['new\_password\_confirm'] = array(

'name' => 'new\_confirm',

'id' => 'new\_confirm',

'type' => 'password',

'pattern' => '^.{'.$this->data['min\_password\_length'].'}.\*$',

);

$this->data['user\_id'] = array(

'name' => 'user\_id',

'id' => 'user\_id',

'type' => 'hidden',

'value' => $user->id,

);

// render

$this->\_render\_page('auth/change\_password', $this->data);

}

else

{

$identity = $this->session->userdata('identity');

$change = $this->ion\_auth->change\_password($identity, $this->input->post('old'), $this->input->post('new'));

if ($change)

{

//if the password was successfully changed

$this->session->set\_flashdata('message', $this->ion\_auth->messages());

$this->logout();

}

else

{

$this->session->set\_flashdata('message', $this->ion\_auth->errors());

redirect('auth/change\_password', 'refresh');

}

}

}

// forgot password

function forgot\_password()

{

// setting validation rules by checking whether identity is username or email

if($this->config->item('identity', 'ion\_auth') != 'email' )

{

$this->form\_validation->set\_rules('identity', $this->lang->line('forgot\_password\_identity\_label'), 'required');

}

else

{

$this->form\_validation->set\_rules('identity', $this->lang->line('forgot\_password\_validation\_email\_label'), 'required|valid\_email');

}

if ($this->form\_validation->run() == false)

{

$this->data['type'] = $this->config->item('identity','ion\_auth');

// setup the input

$this->data['identity'] = array('name' => 'identity',

'id' => 'identity',

);

if ( $this->config->item('identity', 'ion\_auth') != 'email' ){

$this->data['identity\_label'] = $this->lang->line('forgot\_password\_identity\_label');

}

else

{

$this->data['identity\_label'] = $this->lang->line('forgot\_password\_email\_identity\_label');

}

// set any errors and display the form

$this->data['message'] = (validation\_errors()) ? validation\_errors() : $this->session->flashdata('message');

$this->\_render\_page('auth/forgot\_password', $this->data);

}

else

{

$identity\_column = $this->config->item('identity','ion\_auth');

$identity = $this->ion\_auth->where($identity\_column, $this->input->post('identity'))->users()->row();

if(empty($identity)) {

if($this->config->item('identity', 'ion\_auth') != 'email')

{

$this->ion\_auth->set\_error('forgot\_password\_identity\_not\_found');

}

else

{

$this->ion\_auth->set\_error('forgot\_password\_email\_not\_found');

}

$this->session->set\_flashdata('message', $this->ion\_auth->errors());

redirect("auth/forgot\_password", 'refresh');

}

// run the forgotten password method to email an activation code to the user

$forgotten = $this->ion\_auth->forgotten\_password($identity->{$this->config->item('identity', 'ion\_auth')});

if ($forgotten)

{

// if there were no errors

$this->session->set\_flashdata('message', $this->ion\_auth->messages());

redirect("auth/login", 'refresh'); //we should display a confirmation page here instead of the login page

}

else

{

$this->session->set\_flashdata('message', $this->ion\_auth->errors());

redirect("auth/forgot\_password", 'refresh');

}

}

}

// reset password - final step for forgotten password

public function reset\_password($code = NULL)

{

if (!$code)

{

show\_404();

}

$user = $this->ion\_auth->forgotten\_password\_check($code);

if ($user)

{

// if the code is valid then display the password reset form

$this->form\_validation->set\_rules('new', $this->lang->line('reset\_password\_validation\_new\_password\_label'), 'required|min\_length[' . $this->config->item('min\_password\_length', 'ion\_auth') . ']|max\_length[' . $this->config->item('max\_password\_length', 'ion\_auth') . ']|matches[new\_confirm]');

$this->form\_validation->set\_rules('new\_confirm', $this->lang->line('reset\_password\_validation\_new\_password\_confirm\_label'), 'required');

if ($this->form\_validation->run() == false)

{

// display the form

// set the flash data error message if there is one

$this->data['message'] = (validation\_errors()) ? validation\_errors() : $this->session->flashdata('message');

$this->data['min\_password\_length'] = $this->config->item('min\_password\_length', 'ion\_auth');

$this->data['new\_password'] = array(

'name' => 'new',

'id' => 'new',

'type' => 'password',

'pattern' => '^.{'.$this->data['min\_password\_length'].'}.\*$',

);

$this->data['new\_password\_confirm'] = array(

'name' => 'new\_confirm',

'id' => 'new\_confirm',

'type' => 'password',

'pattern' => '^.{'.$this->data['min\_password\_length'].'}.\*$',

);

$this->data['user\_id'] = array(

'name' => 'user\_id',

'id' => 'user\_id',

'type' => 'hidden',

'value' => $user->id,

);

$this->data['csrf'] = $this->\_get\_csrf\_nonce();

$this->data['code'] = $code;

// render

$this->\_render\_page('auth/reset\_password', $this->data);

}

else

{

// do we have a valid request?

if ($this->\_valid\_csrf\_nonce() === FALSE || $user->id != $this->input->post('user\_id'))

{

// something fishy might be up

$this->ion\_auth->clear\_forgotten\_password\_code($code);

show\_error($this->lang->line('error\_csrf'));

}

else

{

// finally change the password

$identity = $user->{$this->config->item('identity', 'ion\_auth')};

$change = $this->ion\_auth->reset\_password($identity, $this->input->post('new'));

if ($change)

{

// if the password was successfully changed

$this->session->set\_flashdata('message', $this->ion\_auth->messages());

redirect("auth/login", 'refresh');

}

else

{

$this->session->set\_flashdata('message', $this->ion\_auth->errors());

redirect('auth/reset\_password/' . $code, 'refresh');

}

}

}

}

else

{

// if the code is invalid then send them back to the forgot password page

$this->session->set\_flashdata('message', $this->ion\_auth->errors());

redirect("auth/forgot\_password", 'refresh');

}

}

// activate the user

function activate($id, $code=false)

{

if ($code !== false)

{

$activation = $this->ion\_auth->activate($id, $code);

}

else if ($this->ion\_auth->is\_admin())

{

$activation = $this->ion\_auth->activate($id);

}

if ($activation)

{

// redirect them to the auth page

$this->session->set\_flashdata('message', $this->ion\_auth->messages());

redirect("auth", 'refresh');

}

else

{

// redirect them to the forgot password page

$this->session->set\_flashdata('message', $this->ion\_auth->errors());

redirect("auth/forgot\_password", 'refresh');

}

}

// deactivate the user

function deactivate($id = NULL)

{

if (!$this->ion\_auth->logged\_in() || !$this->ion\_auth->is\_admin())

{

// redirect them to the home page because they must be an administrator to view this

return show\_error('You must be an administrator to view this page.');

}

$id = (int) $id;

$this->load->library('form\_validation');

$this->form\_validation->set\_rules('confirm', $this->lang->line('deactivate\_validation\_confirm\_label'), 'required');

$this->form\_validation->set\_rules('id', $this->lang->line('deactivate\_validation\_user\_id\_label'), 'required|alpha\_numeric');

if ($this->form\_validation->run() == FALSE)

{

// insert csrf check

$this->data['csrf'] = $this->\_get\_csrf\_nonce();

$this->data['user'] = $this->ion\_auth->user($id)->row();

$this->\_render\_page('auth/deactivate\_user', $this->data);

}

else

{

// do we really want to deactivate?

if ($this->input->post('confirm') == 'yes')

{

// do we have a valid request?

if ($this->\_valid\_csrf\_nonce() === FALSE || $id != $this->input->post('id'))

{

show\_error($this->lang->line('error\_csrf'));

}

// do we have the right userlevel?

if ($this->ion\_auth->logged\_in() && $this->ion\_auth->is\_admin())

{

$this->ion\_auth->deactivate($id);

}

}

// redirect them back to the auth page

redirect('auth', 'refresh');

}

}

// create a new user

function create\_user()

{

$this->data['title'] = $this->lang->line('create\_user\_heading');

if (!$this->ion\_auth->logged\_in() || !$this->ion\_auth->is\_admin())

{

redirect('auth', 'refresh');

}

$tables = $this->config->item('tables','ion\_auth');

$identity\_column = $this->config->item('identity','ion\_auth');

$this->data['identity\_column'] = $identity\_column;

// validate form input

$this->form\_validation->set\_rules('first\_name', $this->lang->line('create\_user\_validation\_fname\_label'), 'required');

$this->form\_validation->set\_rules('last\_name', $this->lang->line('create\_user\_validation\_lname\_label'), 'required');

if($identity\_column!=='email')

{

$this->form\_validation->set\_rules('identity',$this->lang->line('create\_user\_validation\_identity\_label'),'required|is\_unique['.$tables['users'].'.'.$identity\_column.']');

$this->form\_validation->set\_rules('email', $this->lang->line('create\_user\_validation\_email\_label'), 'required|valid\_email');

}

else

{

$this->form\_validation->set\_rules('email', $this->lang->line('create\_user\_validation\_email\_label'), 'required|valid\_email|is\_unique[' . $tables['users'] . '.email]');

}

$this->form\_validation->set\_rules('phone', $this->lang->line('create\_user\_validation\_phone\_label'), 'trim');

$this->form\_validation->set\_rules('company', $this->lang->line('create\_user\_validation\_company\_label'), 'trim');

$this->form\_validation->set\_rules('password', $this->lang->line('create\_user\_validation\_password\_label'), 'required|min\_length[' . $this->config->item('min\_password\_length', 'ion\_auth') . ']|max\_length[' . $this->config->item('max\_password\_length', 'ion\_auth') . ']|matches[password\_confirm]');

$this->form\_validation->set\_rules('password\_confirm', $this->lang->line('create\_user\_validation\_password\_confirm\_label'), 'required');

if ($this->form\_validation->run() == true)

{

$email = strtolower($this->input->post('email'));

$identity = ($identity\_column==='email') ? $email : $this->input->post('identity');

$password = $this->input->post('password');

$additional\_data = array(

'first\_name' => $this->input->post('first\_name'),

'last\_name' => $this->input->post('last\_name'),

'company' => $this->input->post('company'),

'phone' => $this->input->post('phone'),

);

}

if ($this->form\_validation->run() == true && $this->ion\_auth->register($identity, $password, $email, $additional\_data))

{

// check to see if we are creating the user

// redirect them back to the admin page

$this->session->set\_flashdata('message', $this->ion\_auth->messages());

redirect("auth", 'refresh');

}

else

{

// display the create user form

// set the flash data error message if there is one

$this->data['message'] = (validation\_errors() ? validation\_errors() : ($this->ion\_auth->errors() ? $this->ion\_auth->errors() : $this->session->flashdata('message')));

$this->data['first\_name'] = array(

'name' => 'first\_name',

'id' => 'first\_name',

'type' => 'text',

'value' => $this->form\_validation->set\_value('first\_name'),

);

$this->data['last\_name'] = array(

'name' => 'last\_name',

'id' => 'last\_name',

'type' => 'text',

'value' => $this->form\_validation->set\_value('last\_name'),

);

$this->data['identity'] = array(

'name' => 'identity',

'id' => 'identity',

'type' => 'text',

'value' => $this->form\_validation->set\_value('identity'),

);

$this->data['email'] = array(

'name' => 'email',

'id' => 'email',

'type' => 'text',

'value' => $this->form\_validation->set\_value('email'),

);

$this->data['company'] = array(

'name' => 'company',

'id' => 'company',

'type' => 'text',

'value' => $this->form\_validation->set\_value('company'),

);

$this->data['phone'] = array(

'name' => 'phone',

'id' => 'phone',

'type' => 'text',

'value' => $this->form\_validation->set\_value('phone'),

);

$this->data['password'] = array(

'name' => 'password',

'id' => 'password',

'type' => 'password',

'value' => $this->form\_validation->set\_value('password'),

);

$this->data['password\_confirm'] = array(

'name' => 'password\_confirm',

'id' => 'password\_confirm',

'type' => 'password',

'value' => $this->form\_validation->set\_value('password\_confirm'),

);

$view['title'] = 'Создание пользователя';

$this->load->view('header', $view);

$this->\_render\_page('auth/create\_user', $this->data);

$this->load->view('footer');

}

}

// edit a user

function edit\_user($id)

{

$this->data['title'] = $this->lang->line('edit\_user\_heading');

if (!$this->ion\_auth->logged\_in() || (!$this->ion\_auth->is\_admin() && !($this->ion\_auth->user()->row()->id == $id)))

{

redirect('auth', 'refresh');

}

$user = $this->ion\_auth->user($id)->row();

$groups=$this->ion\_auth->groups()->result\_array();

$currentGroups = $this->ion\_auth->get\_users\_groups($id)->result();

// validate form input

$this->form\_validation->set\_rules('first\_name', $this->lang->line('edit\_user\_validation\_fname\_label'), 'required');

$this->form\_validation->set\_rules('last\_name', $this->lang->line('edit\_user\_validation\_lname\_label'), 'required');

$this->form\_validation->set\_rules('phone', $this->lang->line('edit\_user\_validation\_phone\_label'), 'required');

$this->form\_validation->set\_rules('company', $this->lang->line('edit\_user\_validation\_company\_label'), 'required');

if (isset($\_POST) && !empty($\_POST))

{

// do we have a valid request?

if ($this->\_valid\_csrf\_nonce() === FALSE || $id != $this->input->post('id'))

{

show\_error($this->lang->line('error\_csrf'));

}

// update the password if it was posted

if ($this->input->post('password'))

{

$this->form\_validation->set\_rules('password', $this->lang->line('edit\_user\_validation\_password\_label'), 'required|min\_length[' . $this->config->item('min\_password\_length', 'ion\_auth') . ']|max\_length[' . $this->config->item('max\_password\_length', 'ion\_auth') . ']|matches[password\_confirm]');

$this->form\_validation->set\_rules('password\_confirm', $this->lang->line('edit\_user\_validation\_password\_confirm\_label'), 'required');

}

if ($this->form\_validation->run() === TRUE)

{

$data = array(

'first\_name' => $this->input->post('first\_name'),

'last\_name' => $this->input->post('last\_name'),

'company' => $this->input->post('company'),

'phone' => $this->input->post('phone'),

);

// update the password if it was posted

if ($this->input->post('password'))

{

$data['password'] = $this->input->post('password');

}

// Only allow updating groups if user is admin

if ($this->ion\_auth->is\_admin())

{

//Update the groups user belongs to

$groupData = $this->input->post('groups');

if (isset($groupData) && !empty($groupData)) {

$this->ion\_auth->remove\_from\_group('', $id);

foreach ($groupData as $grp) {

$this->ion\_auth->add\_to\_group($grp, $id);

}

}

}

// check to see if we are updating the user

if($this->ion\_auth->update($user->id, $data))

{

// redirect them back to the admin page if admin, or to the base url if non admin

$this->session->set\_flashdata('message', $this->ion\_auth->messages() );

if ($this->ion\_auth->is\_admin())

{

redirect('auth', 'refresh');

}

else

{

redirect('/', 'refresh');

}

}

else

{

// redirect them back to the admin page if admin, or to the base url if non admin

$this->session->set\_flashdata('message', $this->ion\_auth->errors() );

if ($this->ion\_auth->is\_admin())

{

redirect('auth', 'refresh');

}

else

{

redirect('/', 'refresh');

}

}

}

}

// display the edit user form

$this->data['csrf'] = $this->\_get\_csrf\_nonce();

// set the flash data error message if there is one

$this->data['message'] = (validation\_errors() ? validation\_errors() : ($this->ion\_auth->errors() ? $this->ion\_auth->errors() : $this->session->flashdata('message')));

// pass the user to the view

$this->data['user'] = $user;

$this->data['groups'] = $groups;

$this->data['currentGroups'] = $currentGroups;

$this->data['first\_name'] = array(

'name' => 'first\_name',

'id' => 'first\_name',

'type' => 'text',

'value' => $this->form\_validation->set\_value('first\_name', $user->first\_name),

);

$this->data['last\_name'] = array(

'name' => 'last\_name',

'id' => 'last\_name',

'type' => 'text',

'value' => $this->form\_validation->set\_value('last\_name', $user->last\_name),

);

$this->data['company'] = array(

'name' => 'company',

'id' => 'company',

'type' => 'text',

'value' => $this->form\_validation->set\_value('company', $user->company),

);

$this->data['phone'] = array(

'name' => 'phone',

'id' => 'phone',

'type' => 'text',

'value' => $this->form\_validation->set\_value('phone', $user->phone),

);

$this->data['password'] = array(

'name' => 'password',

'id' => 'password',

'type' => 'password'

);

$this->data['password\_confirm'] = array(

'name' => 'password\_confirm',

'id' => 'password\_confirm',

'type' => 'password'

);

$view['title'] = 'Редактировать пользователя';

$this->load->view('header', $view);

$this->\_render\_page('auth/edit\_user', $this->data);

$this->load->view('footer');

}

// create a new group

function create\_group()

{

$this->data['title'] = $this->lang->line('create\_group\_title');

if (!$this->ion\_auth->logged\_in() || !$this->ion\_auth->is\_admin())

{

redirect('auth', 'refresh');

// validate form input

$this->form\_validation->set\_rules('group\_name', $this->lang->line('create\_group\_validation\_name\_label'), 'required|alpha\_dash');

if ($this->form\_validation->run() == TRUE)

{

$new\_group\_id = $this->ion\_auth->create\_group($this->input->post('group\_name'), $this->input->post('description'));

if($new\_group\_id)

{

// check to see if we are creating the group

// redirect them back to the admin page

$this->session->set\_flashdata('message', $this->ion\_auth->messages());

redirect("auth", 'refresh');

}

}

else

{

// display the create group form

// set the flash data error message if there is one

$this->data['message'] = (validation\_errors() ? validation\_errors() : ($this->ion\_auth->errors() ? $this->ion\_auth->errors() : $this->session->flashdata('message')));

$this->data['group\_name'] = array(

'name' => 'group\_name',

'id' => 'group\_name',

'type' => 'text',

'value' => $this->form\_validation->set\_value('group\_name'),

);

$this->data['description'] = array(

'name' => 'description',

'id' => 'description',

'type' => 'text',

'value' => $this->form\_validation->set\_value('description'),

);

$view['title'] = 'Создать группу';

$this->load->view('header', $view);

$this->\_render\_page('auth/create\_group', $this->data);

$this->load->view('footer');

}

}

// edit a group

function edit\_group($id)

{

// bail if no group id given

if(!$id || empty($id))

{

redirect('auth', 'refresh');

}

$this->data['title'] = $this->lang->line('edit\_group\_title');

if (!$this->ion\_auth->logged\_in() || !$this->ion\_auth->is\_admin())

{

redirect('auth', 'refresh');

}

$group = $this->ion\_auth->group($id)->row();

// validate form input

$this->form\_validation->set\_rules('group\_name', $this->lang->line('edit\_group\_validation\_name\_label'), 'required|alpha\_dash');

if (isset($\_POST) && !empty($\_POST))

{

if ($this->form\_validation->run() === TRUE)

{

$group\_update = $this->ion\_auth->update\_group($id, $\_POST['group\_name'], $\_POST['group\_description']);

if($group\_update)

{

$this->session->set\_flashdata('message', $this->lang->line('edit\_group\_saved'));

}

else

{

$this->session->set\_flashdata('message', $this->ion\_auth->errors());

}

redirect("auth", 'refresh');

}

}

// set the flash data error message if there is one

$this->data['message'] = (validation\_errors() ? validation\_errors() : ($this->ion\_auth->errors() ? $this->ion\_auth->errors() : $this->session->flashdata('message')));

// pass the user to the view

$this->data['group'] = $group;

$readonly = $this->config->item('admin\_group', 'ion\_auth') === $group->name ? 'readonly' : '';

$this->data['group\_name'] = array(

'name' => 'group\_name',

'id' => 'group\_name',

'type' => 'text',

'value' => $this->form\_validation->set\_value('group\_name', $group->name),

$readonly => $readonly,

);

$this->data['group\_description'] = array(

'name' => 'group\_description',

'id' => 'group\_description',

'type' => 'text',

'value' => $this->form\_validation->set\_value('group\_description', $group->description),

);

$view['title'] = 'Редактировать группу';

$this->load->view('header', $view);

$this->\_render\_page('auth/edit\_group', $this->data);

$this->load->view('footer');

}

function \_get\_csrf\_nonce()

{

$this->load->helper('string');

$key = random\_string('alnum', 8);

$value = random\_string('alnum', 20);

$this->session->set\_flashdata('csrfkey', $key);

$this->session->set\_flashdata('csrfvalue', $value);

return array($key => $value);

}

function \_valid\_csrf\_nonce()

{

if ($this->input->post($this->session->flashdata('csrfkey')) !== FALSE &&

$this->input->post($this->session->flashdata('csrfkey')) == $this->session->flashdata('csrfvalue'))

{

return TRUE;

}

else

{

return FALSE;

}

}

function \_render\_page($view, $data=null, $returnhtml=false)//I think this makes more sense

{

$this->viewdata = (empty($data)) ? $this->data: $data;

$view\_html = $this->load->view($view, $this->viewdata, $returnhtml);

if ($returnhtml) return $view\_html;//This will return html on 3rd argument being true

}

Children.php

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Children extends CI\_Controller

{

public $limit = 5;

public function \_\_construct()

{

parent::\_\_construct();

if ( ! $this->ion\_auth->logged\_in())

{

redirect('auth/login');

exit;

}

$this->load->model('children\_groups\_model', 'children\_groups');

}

public function report()

{

$data = $this->children->get\_report\_data();

$cols = count($data);

$range = $cols + 1;

$this->load->library('excel');

$this->excel->setActiveSheetIndex(0);

//name the worksheet

$this->excel->getActiveSheet()->setTitle('Список детей');

//set cell A1 content with some text

$this->excel->getActiveSheet()->setCellValue('A1', 'Ф.И.О. ребенка');

$this->excel->getActiveSheet()->setCellValue('B1', 'Дата рождения');

$this->excel->getActiveSheet()->setCellValue('C1', 'Адрес');

$this->excel->getActiveSheet()->setCellValue('D1', 'Дата ПМПК');

$this->excel->getActiveSheet()->setCellValue('E1', '№ протокола');

$this->excel->getActiveSheet()->setCellValue('F1', '№ группы');

$this->excel->getActiveSheet()->getStyle('A1:F1')->getFont()->setBold(true);

$this->excel->getActiveSheet()->getStyle('A1:F' . $range)->getAlignment()->setHorizontal(PHPExcel\_Style\_Alignment::HORIZONTAL\_CENTER);

$this->excel->getActiveSheet()->getStyle('A1:F1')->applyFromArray(

[

'fill' => [

'type' => PHPExcel\_Style\_Fill::FILL\_SOLID,

'color' => ['rgb' => 'D1D1D1']

]

]

);

foreach (range(0, $cols) as $col) {

$this->excel->getActiveSheet()->getColumnDimensionByColumn($col)->setAutoSize(true);

}

for ($i = 0; $i < count($data); $i++)

{

$cell = $i + 2;

$this->excel->getActiveSheet()->setCellValue('A' . $cell, $data[$i]->full\_name);

$this->excel->getActiveSheet()->setCellValue('B' . $cell, $data[$i]->date);

$this->excel->getActiveSheet()->setCellValue('C' . $cell, $data[$i]->address);

$this->excel->getActiveSheet()->setCellValue('D' . $cell, $data[$i]->date\_PMPC);

$this->excel->getActiveSheet()->setCellValue('E' . $cell, $data[$i]->protocol\_number);

$this->excel->getActiveSheet()->setCellValue('F' . $cell, $data[$i]->group\_number);

}

$filename='children\_report.xls'; //save our workbook as this file name

header('Content-Type: application/vnd.ms-excel'); //mime type

header('Content-Disposition: attachment;filename="'.$filename.'"'); //tell browser what's the file name

header('Cache-Control: max-age=0'); //no cache

$objWriter = PHPExcel\_IOFactory::createWriter($this->excel, 'Excel5');

$objWriter->save('php://output');

}

public function index()

{

$config = [

'base\_url' => '/children/index/',

'total\_rows' => $this->children->count(),

'per\_page' => $this->limit,

'use\_page\_numbers' => TRUE

];

$this->pagination->initialize($config);

$data['children'] = $this->children->get\_all($this->limit);

$data['pagination'] = $this->pagination->create\_links();

$view['title'] = 'Список детей';

$this->load->view('header', $view);

$this->load->view('children/index', $data);

$this->load->view('footer');

}

public function search()

{

$data['children'] = $this->children->search($this->input->get('q'));

$data['pagination'] = '';

$view['title'] = 'Результат поиска - Список детей';

$this->load->view('header', $view);

$this->load->view('children/index', $data);

$this->load->view('footer');

}

public function ajax\_search()

{

$data['children'] = $this->children->ajax\_search($this->input->get('q'));

$data['controller'] = 'children';

$this->load->view('ajax\_search', $data);

}

public function edit($id)

{

if ($this->form\_validation->run('children') == FALSE)

{

$data['error'] = '';

$data['groups'] = $this->children\_groups->get\_all();

$data['child'] = $this->children->get\_one($id);

$view['title'] = 'Редактирование - Список детей';

$this->load->view('header', $view);

$this->load->view('children/edit', $data);

$this->load->view('footer');

}

else

{

$config = [

'upload\_path' => './uploads/',

'allowed\_types' => 'gif|jpg|png',

'max\_size' => 4000

];

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('photo'))

{

if ( ! empty($\_FILES['photo']['name']))

{

$data['error'] = $this->upload->display\_errors();

$data['groups'] = $this->children\_groups->get\_all();

$data['child'] = $this->children->get\_one($id);

$view['title'] = 'Редактирование - Список детей';

$this->load->view('header', $view);

$this->load->view('children/edit', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('full\_name, date, address, date\_PMPC, protocol\_number, group\_number');

$this->children->edit($id, $data);

header('Location: /children/');

exit;

}

}

else {

$data = $this->generic->get\_post('full\_name, date, address, date\_PMPC, protocol\_number, group\_number');

$data['photo'] = $this->upload->data('file\_name');

copy('./uploads/' . $data['photo'], './uploads/40x40/' . $data['photo']);

$config = [

'image\_library' => 'gd2',

'source\_image' => './uploads/40x40/' . $data['photo'],

'maintain\_ratio' => FALSE,

'width' => 40,

'height' => 40,

];

$this->load->library('image\_lib', $config);

if ( ! $this->image\_lib->resize())

{

echo $this->image\_lib->display\_errors();

}

else

{

$child\_old = $this->children->get\_photo($id);

if ($child\_old->photo)

{

unlink('./uploads/' . $child\_old->photo);

unlink('./uploads/40x40/' . $child\_old->photo);

}

$this->children->edit($id, $data);

header('Location: /children/');

exit;

}

}

}

}

public function delete($id)

{

$child\_old = $this->children->get\_photo($id);

if ($child\_old->photo)

{

unlink('./uploads/' . $child\_old->photo);

unlink('./uploads/40x40/' . $child\_old->photo);

if ($this->children->delete($id))

{

header('Location: /children/');

exit;

}

}

else

{

if ($this->children->delete($id))

{

header('Location: /children/');

exit;

}

}

}

public function create()

{

//var\_dump($\_POST);die;

if ($this->form\_validation->run('children') == FALSE)

{

$data['error'] = '';

$data['groups'] = $this->children\_groups->get\_all();

$view['title'] = 'Создание - Список детей';

$this->load->view('header', $view);

$this->load->view('children/add', $data);

$this->load->view('footer');

}

else

{

$config = [

'upload\_path' => './uploads/',

'allowed\_types' => 'gif|jpg|png|jpeg',

'max\_size' => 4000

];

$this->load->library('upload', $config);

if ( ! $this->upload->do\_upload('photo'))

{

if ( ! empty($\_FILES['photo']['name']))

{

$data['error'] = $this->upload->display\_errors();

$data['groups'] = $this->children\_groups->get\_all();

$view['title'] = 'Создание - Список детей';

$this->load->view('header', $view);

$this->load->view('children/add', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('full\_name, date, address, date\_PMPC, protocol\_number, group\_number');

if ($id = $this->children->create($data))

{

header('Location: /children/edit/' . $id);

exit;

}

}

}

else

{

$data = $this->generic->get\_post('full\_name, date, address, date\_PMPC, protocol\_number, group\_number');

$data['photo'] = $this->upload->data('file\_name');

copy('./uploads/' . $data['photo'], './uploads/40x40/' . $data['photo']);

$config = [

'image\_library' => 'gd2',

'source\_image' => './uploads/40x40/' . $data['photo'],

'maintain\_ratio' => FALSE,

'width' => 40,

'height' => 40,

];

$this->load->library('image\_lib', $config);

if ( ! $this->image\_lib->resize())

{

echo $this->image\_lib->display\_errors();

}

else {

if ($id = $this->children->create($data)) {

header('Location: /children/edit/' . $id);

exit;

}

}

}

}

}

}

Children\_group.php

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Children\_groups extends CI\_Controller

{

public function \_\_construct()

{

parent::\_\_construct();

$this->load->model('children\_groups\_model', 'children\_groups');

}

public function index()

{

$data['groups'] = $this->children\_groups->get\_all();

$view['title'] = 'Список групп';

$this->load->view('header', $view);

$this->load->view('children\_groups/index', $data);

$this->load->view('footer');

}

public function create()

{

if ($this->form\_validation->run('children\_groups') == FALSE)

{

$view['title'] = 'Добавить группу';

$this->load->view('header', $view);

$this->load->view('children\_groups/add');

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('name');

if ($this->children\_groups->create($data))

{

redirect('/children\_groups');

exit;

}

}

}

public function edit($id)

{

if ($this->form\_validation->run('children\_groups') == FALSE)

{

$data['group'] = $this->children\_groups->get\_one($id);

$view['title'] = 'Редактирование - Группы';

$this->load->view('header', $view);

$this->load->view('children\_groups/edit', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('name');

if ($this->children\_groups->edit($id, $data))

{

redirect('/children\_groups');

exit;

}

}

}

public function delete($id)

{

if ($this->children\_groups->delete($id))

{

redirect('/children\_groups');

exit;

}

}

}

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Classes\_schedule extends CI\_Controller

{

public function \_\_construct()

{

parent::\_\_construct();

if ( ! $this->ion\_auth->logged\_in())

{

redirect('auth/login');

exit;

}

$this->load->model('classes\_schedule\_model', 'classes\_schedule');

$this->load->model('children\_groups\_model', 'groups');

}

public function index()

{

$tmp = $this->classes\_schedule->get\_all();

$data = [];

foreach ($tmp as $row)

{

$data['classes'][$row->day][] = $row;

}

$view['title'] = 'Расписание занятий';

$this->load->view('header', $view);

$this->load->view('classes\_schedule/index', $data);

$this->load->view('footer');

}

public function create()

{

if ($this->form\_validation->run('classes\_schedule') == FALSE)

{

$data['children'] = $this->children->get\_children\_full\_names();

$data['groups'] = $this->groups->get\_all();

$view['title'] = 'Создание - Расписание занятий';

$this->load->view('header', $view);

$this->load->view('classes\_schedule/add', $data);

$this->load->view('footer');

}

else

{

if ($this->input->post('children\_id'))

{

$data = $this->generic->get\_post('day, time, children\_id');

}

else

{

$data = $this->generic->get\_post('day, time, group\_number');

}

if ($this->classes\_schedule->create($data))

{

header('Location: /classes\_schedule');

exit;

}

}

}

public function edit($id)

{

if ($this->form\_validation->run('classes\_schedule') == FALSE)

{

$data['class'] = $this->classes\_schedule->get\_one($id);

$data['children'] = $this->children->get\_children\_full\_names();

$data['groups'] = $this->groups->get\_all();

$view['title'] = 'Редактирование - Расписание занятий';

$this->load->view('header', $view);

$this->load->view('classes\_schedule/edit', $data);

$this->load->view('footer');

}

else

{

if ($this->input->post('children\_id'))

{

$data = $this->generic->get\_post('day, time, children\_id');

}

else

{

$data = $this->generic->get\_post('day, time, group\_number');

}

if ($this->classes\_schedule->edit($id, $data))

{

header('Location: /classes\_schedule');

exit;

}

}

}

public function delete($id)

{

if ($this->classes\_schedule->delete($id))

{

header('Location: /classes\_schedule');

exit;

}

}

}<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Home extends CI\_Controller {

public function index()

{

if ($this->ion\_auth->logged\_in())

{

$view['title'] = 'Logoped';

$this->load->view('header', $view);

$this->load->view('index');

$this->load->view('footer');

}

else

{

redirect('/auth/login');

exit;

}

}

}

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Individual\_card extends CI\_Controller

{

public $limit = 5;

public function \_\_construct()

{

parent::\_\_construct();

if ( ! $this->ion\_auth->logged\_in())

{

redirect('auth/login');

exit;

}

$this->load->model('individual\_card\_model', 'card');

}

public function search()

{

$data['children'] = $this->card->search($this->input->get('q'));

$data['pagination'] = '';

$view['title'] = 'Индивидуальные карты развития';

$this->load->view('header', $view);

$this->load->view('individual\_card/index', $data);

$this->load->view('footer');

}

public function ajax\_search()

{

$data['children'] = $this->card->ajax\_search($this->input->get('q'));

$data['controller'] = 'individual\_card';

$this->load->view('ajax\_search', $data);

}

public function index()

{

$config = [

'base\_url' => '/individual\_card/index/',

'total\_rows' => count($this->card->get\_added\_children('')),

'per\_page' => $this->limit,

'use\_page\_numbers' => TRUE

];

$this->pagination->initialize($config);

$data['children'] = $this->card->get\_added\_children($this->limit);

$data['pagination'] = $this->pagination->create\_links();

$view['title'] = 'Индивидуальные карты развития';

$this->load->view('header', $view);

$this->load->view('individual\_card/index', $data);

$this->load->view('footer');

}

public function view($children\_id)

{

$data['cards'] = $this->card->get\_child\_cards($children\_id);

$data['child'] = $this->children->get\_child\_full\_name($children\_id);

$view['title'] = 'Индивидуальная карта развития';

$this->load->view('header', $view);

$this->load->view('individual\_card/view', $data);

$this->load->view('footer');

}

public function edit($id)

{

if ($this->form\_validation->run('individual\_card') == FALSE)

{

$data['card'] = $this->card->get\_one($id);

$view['title'] = 'Редактирование - Индивидуальная карта развития';

$this->load->view('header', $view);

$this->load->view('individual\_card/edit', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('children\_id, is\_beginning, pronunciation, syllable\_word\_structure, motility, color\_perception, spatial\_perception, eyes\_count\_operations, items\_compare');

$children\_id = $this->input->post('children\_id');

if ($this->card->edit($id, $data))

{

redirect('/individual\_card/view/' . $children\_id);

exit;

}

}

}

public function create()

{

if ($this->form\_validation->run('individual\_card') == FALSE)

{

$data['children'] = $this->card->get\_not\_added\_children();

$view['title'] = 'Создание - Индивидуальная карта развития';

$this->load->view('header', $view);

$this->load->view('individual\_card/add', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('children\_id, is\_beginning, pronunciation, syllable\_word\_structure, motility, color\_perception, spatial\_perception, eyes\_count\_operations, items\_compare');

$card = $this->card->check\_period($data['children\_id']);

if ($card) {

if ($data['is\_beginning'] == $card->is\_beginning)

{

//exception

echo 'На этот период, для этого ребенка запись уже есть, пожалуйста, выберите на другой';

$data['children'] = $this->card->get\_not\_added\_children();

$view['title'] = 'Создание - Индивидуальная карта развития';

$this->load->view('header', $view);

$this->load->view('individual\_card/add', $data);

$this->load->view('footer');

}

}

else

{

if ($this->card->create($data))

{

header('Location: /individual\_card');

exit;

}

}

}

}

public function delete($id)

{

if ($this->card->delete($id))

{

redirect('/individual\_card/');die;

}

}

public function delete\_all($children\_id)

{

if ($this->card->delete\_all($children\_id))

{

redirect('/individual\_card/');

exit;

}

}

}

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Individual\_plan extends CI\_Controller

{

public $limit = 5;

public function \_\_construct()

{

parent::\_\_construct();

if ( ! $this->ion\_auth->logged\_in())

{

redirect('auth/login');

exit;

}

$this->load->model('individual\_plan\_model', 'plan');

$this->load->model('sounds\_model', 'sounds');

}

public function search()

{

$data['plans'] = $this->plan->search($this->input->get('q'));

$data['pagination'] = '';

$view['title'] = 'Индивидуальный план развития';

$this->load->view('header', $view);

$this->load->view('individual\_plan/index', $data);

$this->load->view('footer');

}

public function ajax\_search()

{

$data['children'] = $this->plan->ajax\_search($this->input->get('q'));

$data['controller'] = 'individual\_plan';

$this->load->view('ajax\_search', $data);

}

public function index()

{

$config = [

'base\_url' => '/individual\_plan/index/',

'total\_rows' => $this->plan->count(),

'per\_page' => $this->limit,

'use\_page\_numbers' => TRUE

];

$this->pagination->initialize($config);

$data['plans'] = $this->plan->get\_added\_children($this->limit);

$data['pagination'] = $this->pagination->create\_links();

$view['title'] = 'Индивидуальный план развития';

$this->load->view('header', $view);

$this->load->view('individual\_plan/index', $data);

$this->load->view('footer');

}

public function create()

{

if ($this->form\_validation->run('individual\_plan') == FALSE)

{

$data['children'] = $this->plan->get\_not\_added\_children();

$data['sounds'] = $this->sounds->get\_plan\_sounds();

$view['title'] = 'Создание - Индивидуальный план работы';

$this->load->view('header', $view);

$this->load->view('individual\_plan/add', $data);

$this->load->view('footer');

}

else

{

$sounds = $this->plan->prepare\_addition($this->input->post());

$insert\_id = $this->plan->create($this->plan->plan\_data);

foreach ($sounds as &$item) {

$item['individual\_plan\_id'] = $insert\_id;

}

if ($this->plan->insert\_sounds($sounds))

{

header('Location: /individual\_plan');

exit;

}

}

}

public function edit($id)

{

if ($this->form\_validation->run('individual\_plan') == FALSE)

{

$tmp = $this->plan->get\_one($id);

$data['plan'] = $this->plan->convert\_data\_row($tmp);

$data['sounds'] = $this->sounds->get\_plan\_sounds();

$view['title'] = 'Редактирование - Индивидуальный план работы';

$this->load->view('header', $view);

$this->load->view('individual\_plan/edit', $data);

$this->load->view('footer');

}

else

{

$sounds = $this->plan->prepare\_addition($this->input->post());

$this->plan->edit($this->plan->plan\_id, $this->plan->plan\_data);

if ($this->plan->edit\_sounds($this->plan->plan\_id, $sounds))

{

header('Location: /individual\_plan/');

exit;

}

}

}

public function delete($id)

{

if ($this->plan->delete($id))

{

header('Location: /individual\_plan/');

exit;

}

}

public function report()

{

$tmp = $this->plan->get\_one($id);

$data['plan'] = $this->plan->convert\_data\_row($tmp);

$data['sounds'] = $this->sounds->get\_plan\_sounds();

}

}<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Speech\_card extends CI\_Controller

{

public $limit = 5;

public function \_\_construct()

{

parent::\_\_construct();

if ( ! $this->ion\_auth->logged\_in())

{

redirect('auth/login');

exit;

}

$this->load->model('speech\_card\_model', 'speech\_card');

}

public function search()

{

$data['cards'] = $this->speech\_card->search($this->input->get('q'));

$data['pagination'] = '';

$view['title'] = 'Карта речевого развития';

$this->load->view('header', $view);

$this->load->view('speech\_card/index', $data);

$this->load->view('footer');

}

public function ajax\_search()

{

$data['children'] = $this->speech\_card->search($this->input->get('q'));

$data['controller'] = 'speech\_card';

$this->load->view('ajax\_search', $data);

}

public function index()

{

$config = [

'base\_url' => '/speech\_card/index/',

'total\_rows' => $this->speech\_card->count(),

'per\_page' => $this->limit,

'use\_page\_numbers' => TRUE

];

$this->pagination->initialize($config);

$data['cards'] = $this->speech\_card->get\_added\_children($this->limit);

$data['pagination'] = $this->pagination->create\_links();

$view['title'] = 'Карта речевого развития';

$this->load->view('header', $view);

$this->load->view('speech\_card/index', $data);

$this->load->view('footer');

}

public function create()

{

if ($this->form\_validation->run('speech\_card') == FALSE)

{

$data['children'] = $this->speech\_card->get\_not\_added\_children();

$view['title'] = 'Создание - Карта речевого развития';

$this->load->view('header', $view);

$this->load->view('speech\_card/add', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('children\_id, peu\_number, enrollment\_date, parent\_complaints, native\_language, motility\_state, hearing, general\_development, attention, efficiency, memory, voice, timbre, breath, diction, vocal\_cords, teeth, lips, tongue, movement, bite');

if ($this->speech\_card->create($data))

{

redirect('/speech\_card');

exit;

}

}

}

public function edit($id)

{

if ($this->form\_validation->run('speech\_card') == FALSE)

{

$data['card'] = $this->speech\_card->get\_one($id);

$view['title'] = 'Редактирование - Карта речевого развития';

$this->load->view('header', $view);

$this->load->view('speech\_card/edit', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('children\_id, peu\_number, enrollment\_date, parent\_complaints, native\_language, motility\_state, hearing, general\_development, attention, efficiency, memory, voice, timbre, breath, diction, vocal\_cords, teeth, lips, tongue, movement, bite');

if ($this->speech\_card->edit($id, $data))

{

header('Location: /speech\_card');

exit;

}

}

}

public function delete($id)

{

if ($this->speech\_card->delete($id))

{

redirect('/speech\_card');

exit;

}

}

}

<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Speech\_screen extends CI\_Controller

{

public $limit = 5;

public function \_\_construct()

{

parent::\_\_construct();

if ( ! $this->ion\_auth->logged\_in())

{

redirect('auth/login');

exit;

}

$this->load->model('speech\_screen\_model', 'screen');

$this->load->model('progress\_marks\_model', 'progress');

$this->load->model('sounds\_model', 'sounds');

}

public function search()

{

$tmp = $this->screen->search($this->input->get('q'));

$data['screens'] = $this->screen->convert\_data($tmp);

$data['sounds'] = $this->sounds->get\_screen\_sounds();

$data['pagination'] = '';

$view['title'] = 'Результаты поиска - Речевой экран';

$this->load->view('header', $view);

$this->load->view('speech\_screen/index', $data);

$this->load->view('footer');

}

public function index()

{

$config = [

'base\_url' => '/speech\_screen/index/',

'total\_rows' => $this->screen->count(),

'per\_page' => $this->limit,

'use\_page\_numbers' => TRUE

];

$this->pagination->initialize($config);

$tmp = $this->screen->get\_all($this->limit);

$data['screens'] = $this->screen->convert\_data($tmp);

$data['sounds'] = $this->sounds->get\_screen\_sounds();

$data['pagination'] = $this->pagination->create\_links();

$view['title'] = 'Речевой экран';

$this->load->view('header', $view);

$this->load->view('speech\_screen/index', $data);

$this->load->view('footer');

}

public function report()

{

$tmp = $this->screen->get\_all();

$screens = $this->screen->convert\_data($tmp);

$sounds = $this->sounds->get\_screen\_sounds();

$cols = count($screens);

$range = $cols + 1;

$this->load->library('excel');

$this->excel->setActiveSheetIndex(0);

//name the worksheet

$this->excel->getActiveSheet()->setTitle('Речевой экран');

//set cell A1 content with some text

$this->excel->getActiveSheet()->setCellValue('A1', 'Ф.И.О. ребенка');

$this->excel->getActiveSheet()->setCellValue('B1', 'ФФ. восприятие');

$this->excel->getActiveSheet()->setCellValue('C1', 'Год обучения');

$this->excel->getActiveSheet()->setCellValue('D1', 'Диагноз');

$col\_letter = 'E';

foreach ($sounds as $sound)

{

$this->excel->getActiveSheet()->setCellValue($col\_letter . '1', $sound->name);

$col\_letter++;

}

$col\_letter = chr(ord($col\_letter) - 1);

$this->excel->getActiveSheet()->getStyle('A1:' . $col\_letter . '1')->getFont()->setBold(true);

$this->excel->getActiveSheet()->getStyle('A1:' . $col\_letter . $range)->getAlignment()->setHorizontal(PHPExcel\_Style\_Alignment::HORIZONTAL\_CENTER);

$this->excel->getActiveSheet()->getStyle('A1:' . $col\_letter . '1')->applyFromArray(

[

'fill' => [

'type' => PHPExcel\_Style\_Fill::FILL\_SOLID,

'color' => ['rgb' => 'D1D1D1']

]

]

);

foreach (range(0, $cols) as $col) {

$this->excel->getActiveSheet()->getColumnDimensionByColumn($col)->setAutoSize(true);

}

for ($i = 0; $i < count($screens); $i++)

{

$cell = $i + 2;

$this->excel->getActiveSheet()->setCellValue('A' . $cell, $screens[$i]['full\_name']);

$this->excel->getActiveSheet()->setCellValue('B' . $cell, $screens[$i]['ff\_perception']);

$this->excel->getActiveSheet()->setCellValue('C' . $cell, $screens[$i]['study\_year']);

$this->excel->getActiveSheet()->setCellValue('D' . $cell, $screens[$i]['diagnosis']);

$col\_letter = 'E';

foreach ($screens[$i]['sounds'] as $sound)

{

$this->excel->getActiveSheet()->setCellValue($col\_letter++ . $cell, $sound['progress\_symbol']);

}

}

$filename='speech\_screen\_report.xls'; //save our workbook as this file name

header('Content-Type: application/vnd.ms-excel'); //mime type

header('Content-Disposition: attachment;filename="'.$filename.'"'); //tell browser what's the file name

header('Cache-Control: max-age=0'); //no cache

$objWriter = PHPExcel\_IOFactory::createWriter($this->excel, 'Excel5');

$objWriter->save('php://output');

}

public function ajax\_search()

{

$data['children'] = $this->screen->ajax\_search($this->input->get('q'));

$data['controller'] = 'speech\_screen';

$this->load->view('ajax\_search', $data);

}

public function create()

{

if ($this->form\_validation->run('speech\_screen') == FALSE)

{

$data['children'] = $this->screen->get\_not\_added\_children();

$data['sounds'] = $this->sounds->get\_screen\_sounds();

$data['progress'] = $this->progress->get\_all();

$view['title'] = 'Создание - Речевой экран';

$this->load->view('header', $view);

$this->load->view('speech\_screen/add', $data);

$this->load->view('footer');

}

else

{

$sounds = $this->screen->prepare\_addition($this->input->post());

$insert\_id = $this->screen->create($this->screen->screen\_data);

foreach ($sounds as &$item) {

$item['speech\_screen\_id'] = $insert\_id;

}

if ($this->screen->insert\_sounds($sounds))

{

header('Location: /speech\_screen');

exit;

}

}

}

public function edit($id)

{

if ($this->form\_validation->run('speech\_screen') == FALSE)

{

$tmp = $this->screen->get\_one($id);

$data['screen'] = $this->screen->convert\_data\_row($tmp);

$data['sounds'] = $this->sounds->get\_screen\_sounds();

$data['progress'] = $this->progress->get\_all();

$view['title'] = 'Редактирование - Речевой экран';

$this->load->view('header', $view);

$this->load->view('speech\_screen/edit', $data);

$this->load->view('footer');

}

else

{

$sounds = $this->screen->prepare\_addition($this->input->post());

$this->screen->edit($this->screen->screen\_id, $this->screen->screen\_data);

if ($this->screen->edit\_sounds($this->screen->screen\_id, $sounds))

{

header('Location: /speech\_screen');

exit;

}

}

}

public function delete($id)

{

if ($this->screen->delete($id))

{

header('Location: /speech\_screen/');

exit;

}

}

}<?php

defined('BASEPATH') OR exit('No direct script access allowed');

class Work\_schedule extends CI\_Controller

{

public function \_\_construct()

{

parent::\_\_construct();

if (!$this->ion\_auth->logged\_in()) {

redirect('auth/login');

exit;

}

$this->load->model('work\_schedule\_model', 'work\_schedule');

}

public function index()

{

$data['schedule'] = $this->work\_schedule->get\_all();

$view['title'] = 'График работы';

$this->load->view('header', $view);

$this->load->view('work\_schedule/index', $data);

$this->load->view('footer');

}

public function edit($id)

{

if ($this->form\_validation->run('work\_schedule') == FALSE)

{

$data['schedule'] = $this->work\_schedule->get\_one($id);

$view['title'] = 'Редактирование - График работы';

$this->load->view('header', $view);

$this->load->view('work\_schedule/edit', $data);

$this->load->view('footer');

}

else

{

$data = $this->generic->get\_post('day, start\_time, end\_time');

if ($this->work\_schedule->edit($id, $data))

{

redirect('/work\_schedule');

exit;

}

}

}

public function delete($id)

{

if ($this->work\_schedule->delete($id))

{

redirect('/work\_schedule');

exit;

}

}

public function create()

{

if ($this->form\_validation->run('work\_schedule') == FALSE)

{

$view['title'] = 'Добавление - График работы';

$this->load->view('header', $view);

$this->load->view('work\_schedule/add');

$this->load->view('footer');

}

else

{

$days = $this->work\_schedule->get\_days();

foreach ($days as $value)

{

if ($value->day == $this->input->post('day'))

{

$data['error'] = 'Такой день уже есть, пожалуйста отредактируйте или выберите другой день';

$view['title'] = 'Добавление - График работы';

$this->load->view('header', $view);

$this->load->view('work\_schedule/add', $data);

$this->load->view('footer');

break;

}

else

{

$data = $this->generic->get\_post('day, start\_time, end\_time');

if ($this->work\_schedule->create($data))

{

}

**РЕЦЕНЗИЯ**

**Для дипломной работы** образовательно-квалификационного уровня «бакалавр», выполненной на тему «Автоматизированная информационная система учета работы учителя-логопеда психолого-медико-педагогической консультации»,

студентом групи КН-12 – Хараманом Владимиром Сергеевичем.

Представленная дипломная работа образовательно-квалификационного уровня «бакалавр» посвящена автоматизации учета-работы учителя-логопеда, автоматизации документаций учителя-логопеда. Учитывая тот факт, что достойных аналогов данного проекта нет, а автоматизировать такого рода работу необходимо как можно быстрее, - дипломная работа Харамана В.С. характеризуется актуальностью.

Содержание работы соответствует выбранной теме. Дипломная работа состоит из введения, двух глав, заключения. В первой главе он раскрывает теоретические основы, и объясняет свою предметную область в доступном виде. Особого внимания в первой главе заслуживает проведенный в работе анализ существующего уровня автоматизации.

Вторая глава дипломной работы посвящена проектированию системы, и ее архитектуре. В том числе хотелось бы отметить исчерпывающую инструкцию пользователя. С помощью нее было очень легко разобраться в интерфейсе приложения, все элементы расположены интуитивно понятно.

В работе также рассматривается проектирование базы данных, хорошо проанализированная ее архитектура и реализация. Видно что база данных была хорошо нормализована.

Положительными чертами работы является системность и последовательность изложения материала, а также применение собственного опыта в сфере разработки.

Дипломная работа образовательно-квалификационного уровня «бакалавр» выполнена в полном объеме, заслуживает оценки «отлично», а ее автору, Хараману Владимиру Сергеевичу, может быть присвоена квалификация бакалавра.

Руководитель компании MakeBeCool Гадашевич Андрей Владимирович.